

CLUSTER/ONE MODEL ATM

Network File Server Version 1

Product Number A-2801

User's Manual

NESTAR



NESTAR SYSTEMS, INCORPORATED

CLUSTER/ONE MODEL A (TM)

NETWORK FILE SERVER Version 1
Product Number A-2801

USER'S MANUAL

DISCLAIMER

NOTICE

Nestar Systems, Incorporated, reserves the right to make changes to any products described herein to improve their functioning or design.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

The information in this document has been carefully reviewed and is believed to be reliable; nevertheless, Nestar Systems, Incorporated makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. The entire risk as to its quality and performance is with the buyer. Nestar Systems, Incorporated software is licensed "as is." The terms and conditions under which programming errors will be corrected are described in the Software License Agreement for the product.

In no event will Nestar Systems, Incorporated be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software, even if Nestar Systems, Incorporated has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

The Regents of the University of California and Apple Computer, Inc. make no warranties, either express or implied, regarding the Apple Pascal computer software package described herein, its merchantability or its fitness for any particular purpose.

This document is licensed under the Software License Agreement for the program described herein (Network File Server - Product Number A-2801).

This document is copyrighted and all rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Nestar Systems, Incorporated.

Copyright 1980, 1981 by Nestar Systems, Incorporated
2585 E. Bayshore, Palo Alto, California, 94303
(415) 493-2223

Publication number LC20-0101-1
All rights reserved.

How to Use This Manual

This is the first revision of the User Manual, corresponding to Version 1.1 of the Network File Server. It is intended for experienced programmers who know the Apple DOS, BASIC, and Apple Pascal programming environments. The first part of this manual is designed as a tutorial for new users. The remainder is a reference manual for experienced users of the Cluster/One Model A Network File Server. New users should see especially:

Chapter 1, An Overview

Chapter 2, A hands-on tutorial for a minimal system

There are also chapters on disk organization (Chap. 3), protection (Chap. 4), and using the Network File Server from DOS (Chap. 6) and Pascal (Chap. 7).

For reference purposes, users should see:

Chapter 5, where individual commands are described in alphabetical order

Chapter 8, for descriptions of Nestar utility programs

Appendix A, a list of error and information messages

Appendix E, a quick reference list of commands and parameters

This edition of the Users Manual incorporates all modifications to the Cluster/One Model A Nestar File Server Version 1.1. Of special interest will be new support for

- (a) Two Winchester disk drives; total capacity of 66 Megabytes, formatted
- (b) New Apple operating systems DOS 3.3 and Pascal 1.1
- (c) DOS virtual disks of variable size
- (d) Electronic office mail and Clock/Calendar options
- (e) New DOS and Pascal user utilities
- (f) Enhanced file server commands
- (g) Multiple file servers on one or more networks

Appendix C provides a checklist of Version 1.1 changes for the former 1.0 user.

We welcome criticisms and suggestions. Forms for reporting

THIS MANUAL

program errors and documentation errors or inadequacies are included near the back of the manual.

Table of Contents

How to Use This Manual.....	ii
Chapter 1 Introduction.....	1-1
1.1 Up to 65 Apples Are Networked Together.....	1-1
1.2 DOS and Pascal Think They're Alone.....	1-1
1.3 Network File Server Commands Control Virtual Diskettes.....	1-2
1.4 Shared Disk System: Floppy and Hard Disk.....	1-2
1.5 Dedicated User-Stations Control I/O Devices.....	1-3
1.6 User-Stations and the Network File Server.....	1-4
Chapter 2 The Minimal System.....	2-1
2.1 Virtual Diskettes.....	2-1
2.2 Virtual Drives.....	2-2
2.3 The Simplest Disk Organization.....	2-2
2.4 Registration and Logon Procedures.....	2-3
2.5 Issuing Direct File Server Commands Under DOS....	2-5
2.6 Initializing a DOS Virtual Diskette.....	2-9
2.7 Issuing File Server Commands From Within a BASIC Program.....	2-10
2.8 Copying a Program Library.....	2-12
2.8.1 Copying from a Cassette Tape.....	2-12
2.8.2 Copying from a Diskette.....	2-13
2.9 Other Features.....	2-13
Chapter 3 Disk Organization and Pathnaming Conventions.....	3-1
3.1 Disk Organization.....	3-1
3.1.1 Disk Units.....	3-1
3.1.2 Files.....	3-1
3.1.3 Directories.....	3-2
3.1.4 Virtual Diskettes.....	3-2
3.1.5 Network Utilities.....	3-2
3.2 Pathnaming Conventions.....	3-3
3.2.1 Pathname: /unitnumber.....	3-3
3.2.2 Pathname: /unitname.....	3-3
3.2.3 Pathname: /unitname/directory.....	3-3
3.2.4 Pathname: /unitname/directory1 /directory2/virtual diskette.....	3-4
3.2.5 Current Default Directory.....	3-4
3.2.6 Passwords.....	3-5
3.2.7 Pathnames--A Formal Definition.....	3-5

CONTENTS

Chapter 4	Protection.....	4-1
4.1	NFS File Protection.....	4-2
4.1.1	User Classes.....	4-2
4.1.2	Passwords and Access Rights.....	4-4
4.1.3	Default Access Rights.....	4-5
4.1.4	The PROTECT Command.....	4-6
4.1.5	The SHOW PROTECTION Command.....	4-6
4.1.6	Preset Passwords.....	4-7
4.1.7	Access Rights Needed for Network Commands.....	4-7
4.1.8	What Happens If You Forget a Password.....	4-8
4.2	Usage Modes and the MOUNT Command.....	4-9
4.3	Locks.....	4-9
4.3.1	LOCK Command.....	4-10
4.3.2	UNLOCK Command.....	4-11
4.3.3	SHOW LOCK(S) Command.....	4-11
Chapter 5	Network File Server Commands.....	5-1
BLOAD.....		5-5
BRUN.....		5-5
BSAVE.....		5-5
CREATE.....		5-9
DELETE.....		5-14
HELP.....		5-15
LIST.....		5-16
LOCK.....		5-19
MOUNT.....		5-21
OFF.....		5-23
PATHNAME.....		5-24
PROTECT.....		5-25
PROTECTION.....		5-27
RENAME.....		5-28
SET DIR.....		5-29
SET.....		5-31
SHOW DATE.....		5-33
SHOW DIR.....		5-34
SHOW LOCK.....		5-35
SHOW [ALL] LOCKS.....		5-36
SHOW [ALL] MOUNTS.....		5-38
SHOW PROTECTION.....		5-41
SHOW STATION.....		5-42
SHOW TYPE.....		5-43
SHOW TYPES.....		5-44
SHOW VOLS.....		5-45
TIMESTAMP.....		5-46
UNLOCK.....		5-47
UNMOUNT.....		5-48

Chapter 6 Using DOS with the Network File Server....6-1

6.1	Size and Number of Sectors.....	6-1
6.2	Drives.....	6-1
6.3	Using Immediate Execution Mode in DOS.....	6-1
6.4	Using File Server Commands within a BASIC Program.....	6-2
6.4.1	Applesoft BASIC.....	6-2
6.4.2	Integer BASIC.....	6-3
6.5	Initializing a Virtual Diskette in DOS.....	6-4
6.6	Creating and INITing a Variable-Sized Virtual Diskette Using VDOS CREATE.....	6-4
6.7	Using DOS From Real Diskettes.....	6-6

Chapter 7 Using Pascal with the Cluster/One System...7-1

7.1	Virtual Diskettes of Variable Size.....	7-1
7.2	Drives.....	7-2
7.3	The CMD Program. Entering File Server Commands From The Keyboard.....	7-3
7.3.1	SHOW DRIVES.....	7-4
7.3.2	SHOW NFS.....	7-5
7.3.3	SHOW SLOTS.....	7-5
7.3.4	SET DRIVE[S].....	7-5
7.3.5	SET NFS.....	7-6
7.3.6	Other CMD Inputs.....	7-6
7.4	Using File Server Commands Within a Pascal Program.....	7-7
7.4.1	CMDUNIT.....	7-8
7.4.2	NETUNIT.....	7-9
7.4.3	A Sample Program.....	7-11
7.5	Multiple File Servers.....	7-12
7.6	Initializing a Virtual Diskette in Pascal.....	7-13

Chapter 8 User Station Utilities.....8-1

8.1	Introduction.....	8-1
8.2	Pascal User Utilities.....	8-3
8.2.1	SETPROFILE Utility Program.....	8-3
8.2.2	NESTAR.START Command Program.....	8-4
8.2.3	FILEINFO Utility Program.....	8-7
8.2.3.1	How to Run FILEINFO.....	8-7
8.2.4	TDELETE Utility Program.....	8-10
8.2.5	NETLINK Program.....	8-11
8.2.5.1	How to Run NETLINK.....	8-11
8.2.6	DIRLIST Utility Program.....	8-13
8.3	DOS User Station Programs.....	8-15
8.3.1	SETPROFILE Utility Program.....	8-15
8.3.2	NESTAR DISK COPY Utility Program.....	8-16
8.3.2	How to Run NESTAR DISK COPY.....	8-16
8.3.3	FILEINFO Utility Program.....	8-18

CONTENTS

8.3.4 FIDMORE.....	8-21
8.3.4.1 How to Run FIDMORE.....	8-21

Appendices

A Error and Information Messages.....	A-1
B The BOOT Program.....	B-1
B.1 Description.....	B-1
B.2 Operation of BOOT.....	B-2
B.2.1 Booting DOS Virtual Diskettes.....	B-3
B.2.2 Booting Pascal Virtual Diskettes.....	B-3
B.2.3 Autobooting at the Individual Station.....	B-4
C Changes to Cluster/One Model A in Release 1.1.....	C-1
D Reading the Station Number from a Program.....	D-1
D.1 Introduction.....	D-1
D.2 Machine Language.....	D-1
D.3 BASIC.....	D-1
D.4 Pascal.....	D-2
D.5 Special Locations in the Network Interface Card..	D-2
E Quick Reference List - File Server Commands.....	E-1
INDEX.....	IN-1

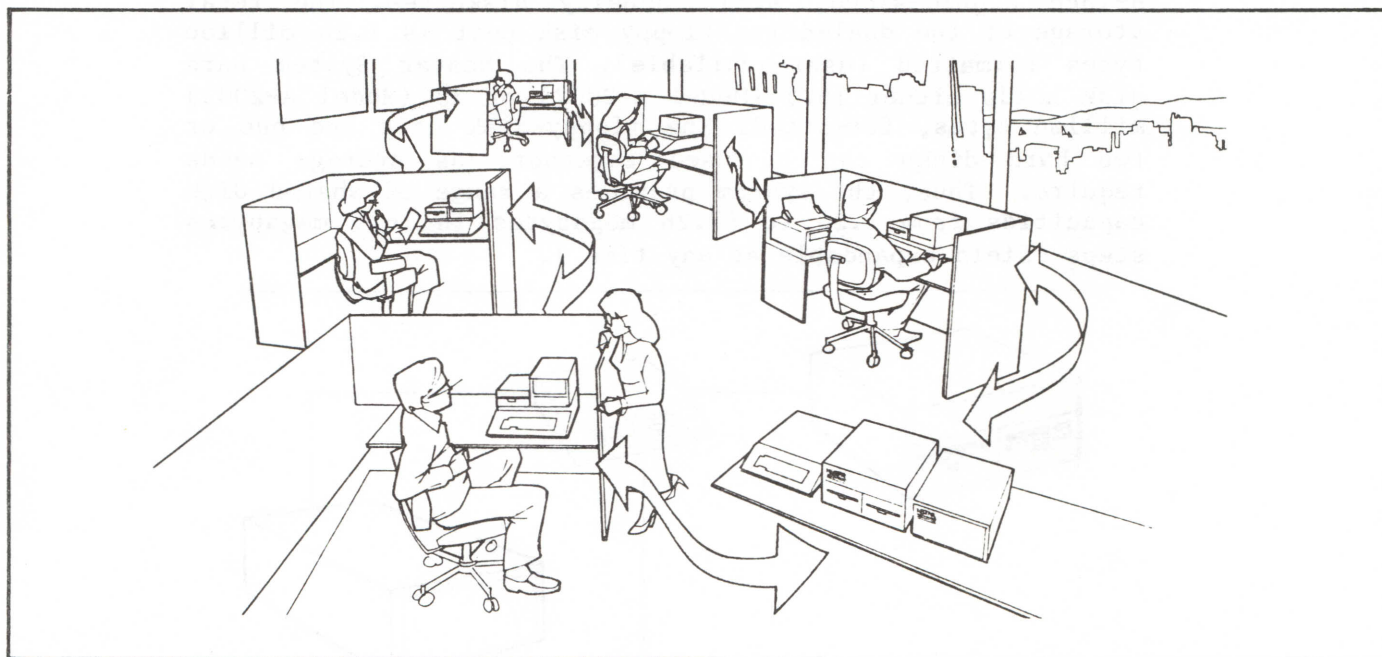
READERS COMMENT FORM

REFERENCE FORM (COMMONLY USED VOLUME/FILE NAMES)

Chapter 1

Introduction1.1 Up to 65 Apples Are Networked Together

The Cluster/One Model A system, developed by Nestar Systems, Incorporated of Palo Alto, California, is a hard-wired network of up to 65 Apple microcomputers which share resources such as a disk storage system. One or more Apples can act as network file-servers, transferring programs and data between the disk storage and the user-station Apples. Except during such disk transfers, the individual user-stations function as independent computers with full storage and computing capabilities. In large part, this manual is concerned with the relationship between the individual user stations and the Apples that act as file servers.

1.2 DOS and Pascal Think They're Alone

To the individual user, the file server disk storage system is a set of virtual diskettes which are indistinguishable from the 5 1/4 inch diskettes and drives on a stand-alone Apple. Almost all programs which run on a stand-alone Apple can run without modification using the Cluster/One Model A Network file server. Apples which have appropriate hardware (e.g. a language card) can run under either the Pascal Operating System or the Disk Operating System (DOS) for Integer and Applesoft BASIC. A Pascal program can have as many as ten virtual diskettes accessible by the program at

INTRO

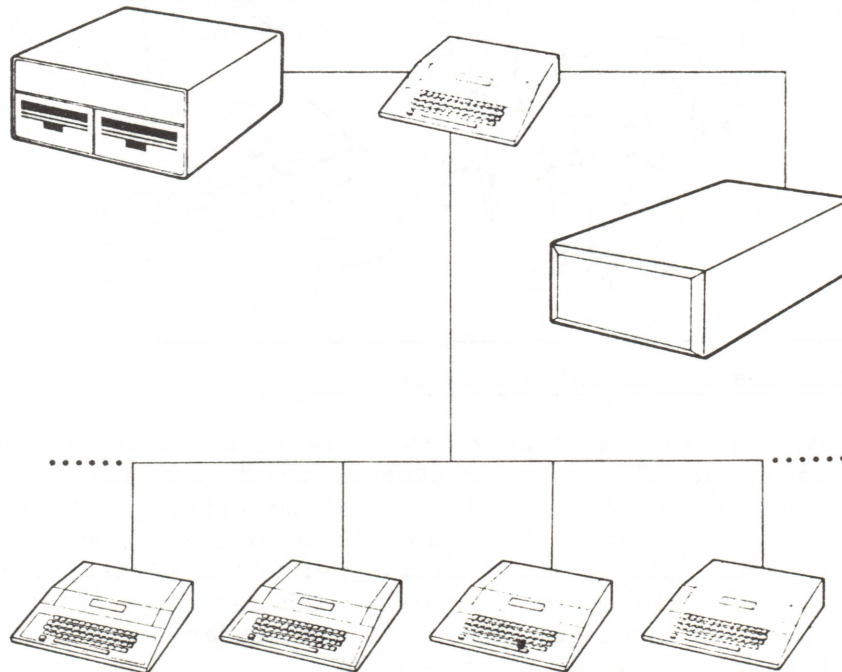
the same time, and a BASIC program can have as many as 255.

1.3 Network File Server Commands Control Virtual Diskettes

Network File Server commands are available for creating, mounting, and protecting virtual diskettes and for displaying the status of the Network File Server directories, mounts, and locks. Once a virtual diskette has been mounted, it is available to either DOS or Pascal for the full range of file operations: the contents of volumes can be listed; programs can be loaded and saved; and data files can be read and written, just as with a real diskette.

1.4 Shared Disk System: Floppy And Hard Disk

The minimum system controlled by the Network File Server uses a dual-drive floppy disk unit (Model A-2001) with 8-inch double-sided, single-density diskettes. The total storage of the dual-drive floppy disk unit is 1.26 million bytes formatted (user-available). The Nestar System hard disk holds either 16.5 (Model A-2002) or 33 (Model A-2003) million bytes, formatted. The floppy-disk unit and one or two hard disks can be used together, as storage needs require. Thus, the system provides a range of shared disk capacities from 1.26 to 67.26 megabytes in 16.5 megabyte steps (field expandable at any time.).

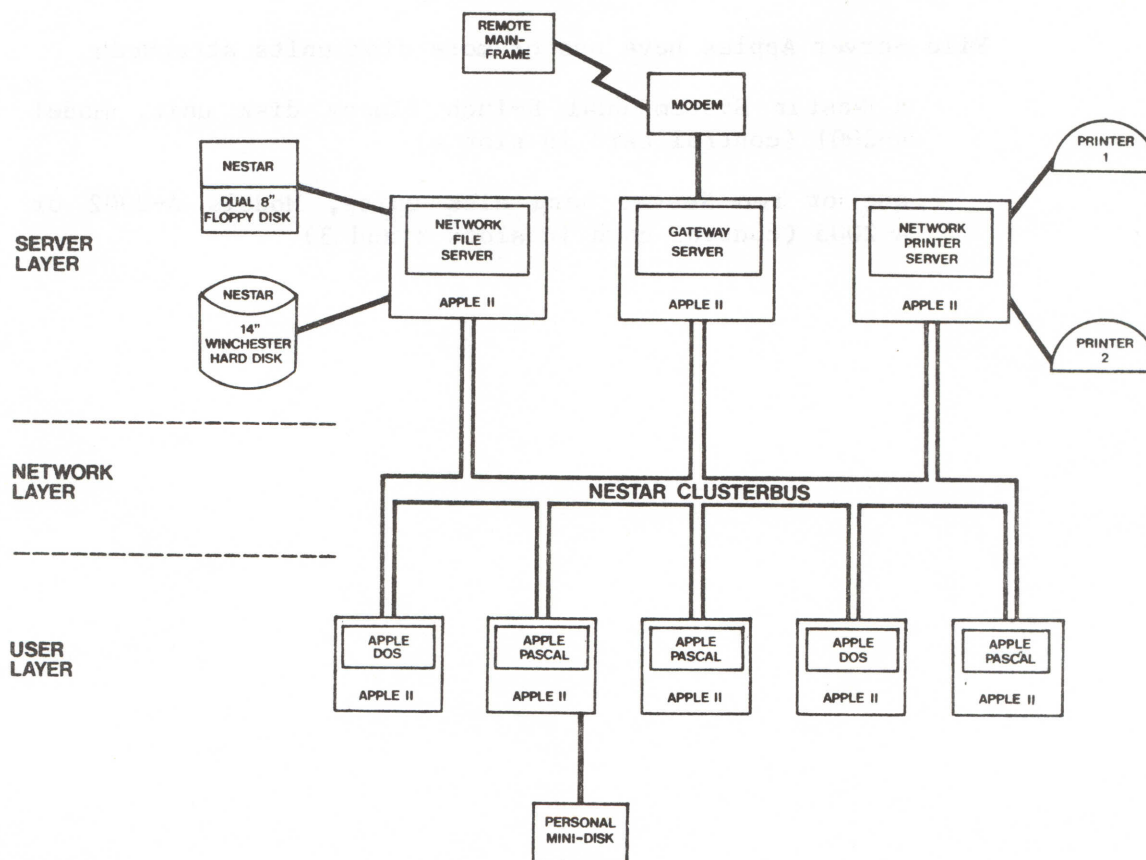


The Nestar "Model A" system uses one dedicated Apple as central mass storage manager which connects with all the user stations through the Nestar Cluster Bus.

1.5 Dedicated User-Stations Control I/O Devices

Each user-station can control I/O devices for its own private use. If a device, say the printer, is to be used by more than one user, then one of the stations (called a "server-station") can listen for and execute requests from others on the network for printer service. Though dedicated in the sense that it listens periodically for requests, the station can perform other tasks that do not interfere with its dedicated function. Examples of services that could be provided by dedicated server-stations and thus made available to more than one user are:

communicating by phone line with a mainframe computer
 communicating by phone line with other Cluster/One networks
 collecting data from a scientific instrument
 interfacing with a phone line to a remote user
 reading stylus positions from a graphics tablet
 formatting and printing text for word processing
 plotting data



1.6 User-Station and the Network File Server

The Apples that function as stations on the Cluster/One Model A system must have a network interface card (Network Systems Product Number A-2701 or A-2702) inserted in a peripheral slot (other than 0). If the Apple is connected to more than one network, additional network card(s) are inserted in other slots. If the user station runs the Pascal Operating System, it must have a 48K memory and a Language Card in slot 0. DOS requires only a 32K memory, or even 16K for some minimum applications. DOS requires a Language Card or ROM card only if both Integer and Floating Point BASIC are to be used.

Apples that function as network file servers must have 64K main memory. The Apple functioning as the first network file server on a network must use a Nestar System A-2702 network interface card; other file server Apples must use A-2703 cards. A-2702 and A-2703 cards use a special program ROM (labeled B1.1) for autobooting the system from the floppy or hard disk. The A-2702 card contains a resistor-termination-pack chip and is addressed \$FE. The A-2703 cards each have a unique address.

File Server Apples have one or more disk units attached:

- a Nestar System dual 8-inch floppy disk unit, Model A-2001 (control card in slot 4)

- one or two Nestar hard disk units, Models A-2002 or A-2003 (control card in slots 2 and 3)

Chapter 2

The Minimal System

The Cluster/One Model A Network File Server provides a sophisticated file system involving complex file-naming conventions and protection procedures. However, in this chapter, which is designed as a hands-on tutorial for the beginning user, it will be treated as a minimal, no-frills system. This chapter will present the framework of the system. Once the fundamental concepts are mastered, the more sophisticated features presented later will be seen as logical extensions.

Another simplification in this tutorial chapter is that the examples will all be for DOS user stations. The relationship of Cluster/One to DOS is similar but not identical to its relationship to the Pascal Operating System. However, once the rules for issuing file server commands under DOS are mastered, the different rules for Pascal will be easily understood. For more details on using Network commands within each operating system, see Chapter 6 for DOS and Chapter 7 for Pascal.

The fundamental concepts of the Cluster/One Model A system are:

- virtual diskettes
- virtual drives
- the file structure
- getting started
- issuing direct file server commands
- the essential file server commands:

CREATE
DELETE
MOUNT
UNMOUNT
LIST
SHOW MOUNTS

- initializing a virtual diskette under DOS
- issuing file server commands from a BASIC program.

2.1 Virtual Diskettes

The objective of the network is to allow each individual user to program with exactly the same DOS, BASIC, or Pascal commands used with a stand-alone Apple, and, in addition, to use as many virtual diskettes as needed. With a stand-alone Apple, a large data-base program requires constant insertion and removal of diskettes in the 5 1/4 inch disk

drives; Nestar provides the same capability by allowing the user to create, delete, mount, and unmount virtual diskettes which reside on the disk units controlled by Network file servers. Each virtual diskette has the same internal structure, controlled by either the DOS or Pascal Operating System, that a real diskette has. Since the formats required by DOS and Pascal are different, DOS and Pascal programs and data cannot be stored on the same virtual diskette.

Even though there can be as many as 64 users all sharing disk units, each individual user still operates as if he were using a stand-alone Apple with his own drives. The only difference is that instead of using a diskette, he issues a Network file server command to create a virtual diskette, and instead of physically inserting a real diskette into a real drive, he issues a Network file server command to mount a virtual diskette on a virtual drive. DOS and Pascal don't know the difference.

The CREATE command allocates disk space which is equivalent to a diskette and is referred to in this manual as either a virtual diskette or a volume. The CREATE command also puts the name and address of the virtual diskette in a NFS directory.

2.2 Virtual Drives

A real disk drive must have a real diskette inserted before it is useful; correspondingly, a virtual drive must have a virtual diskette mounted before it is useful. The MOUNT command establishes the correspondence between a virtual diskette and a virtual drive, so that when DOS (or Pascal) is instructed to read or write on drive n, it will read or write on the virtual diskette corresponding to (MOUNTED on) virtual drive n.

The network interface card can be thought of as a disk controller; however, it can control more than two (virtual) drives. Ten Pascal and 255 DOS drives are available to each user station.

2.3 The Simplest Disk Organization

Each file server floppy disk or hard disk drive is divided into:

- directories
- virtual diskettes
- utility programs
- other files
- unused space

In the simplest case there is just one directory (called a root directory) on a given 8 inch system floppy disk or hard disk. This directory contains the names and addresses of all virtual diskettes and other files stored on that disk unit; it cannot contain names of virtual diskettes or other files stored on other disk units.

To access a particular virtual diskette, the user specifies a pathname which contains:

the name or number of the physical disk unit the virtual diskette is stored on

the name of the virtual diskette itself.

For example, if an NFS file is called LINDA and is stored on the diskette inserted in the left drive of the first system floppy disk unit (unit 1), the pathname is:

/1/LINDA

Obviously, the system diskette could just as easily be inserted in the right hand drive (unit 2) or, for faster access, copied onto a hard disk. Therefore, it is often more convenient to give the name of the diskette rather than the number corresponding to its current location. For example, with each Cluster/One Model A system, Nestar supplies a system diskette called MAIN. Included on MAIN is a virtual diskette called BOOT. To address BOOT, irrespective of where the system disk is physically located, the pathname is:

/MAIN/BOOT

In a slightly more complex case, each root directory can contain the names and addresses of other directories, as well as the names of virtual diskettes. For example, in a typical installation, a root directory may have an entry called USERS. USERS, in turn, is a list of directories, one for each user--e.g. TOM, DICK, HARRY, MILHOUS. The pathname for the NFS directory represented by the name TOM would be /MAIN/USERS/TOM. The directory TOM might contain entries for the virtual diskettes used by Tom. See Section 3.2 for a complete discussion of pathnames.

2.4 Registration and Logon Procedures

The remainder of this chapter is intended as a hands-on tutorial to get you started with the Cluster/One Model A system. Since all installations are different, ask your system manager to register you and to tell you how to begin

MINIMAL SYSTEM

using a user terminal. For readers who do not have access to a user-terminal while reading this chapter, the procedure supplied by Nestar is described.

Ask your system manager to give you the pathname of an NFS virtual volume that has been formatted to boot DOS and that you can use to try out file server commands and create your own private virtual diskette. For purposes of this tutorial, we shall assume that you are given the name:

/MAIN/DOS/3.3

At a user-station, turn the Apple power switch (left, rear) off and then on to ensure a fresh start. (Later you will learn to accomplish the same thing by typing @OFF when your Apple has been set to accept Network commands.) The screen will display:



CLUSTER/ONE MODEL A NETWORK BOOT 1.1
STATION ADDRESS \$nn
SLOT 6 SERVER \$FE

VOLUME TO BOOT: /MAIN/

where nn is the number of your station (in hexadecimal).

Enter the pathname exactly as your system manager gave it to you. You can use the backarrow to backspace to the M of MAIN if your virtual diskette is on some other disk unit. For the Nestar example, type DOS/3.3. The last line will now read:

VOLUME TO BOOT: /MAIN/DOS/3.3

(Note: Terminate all inputs to the Network file server with a carriage return.)

If everything was entered correctly, the speaker will click softly for a few seconds to indicate network transactions and a display similar to the following will appear:

DOS VERSION 3.3	08/25/80
APPLE II PLUS OR ROM CARD	SYSTEM MASTER
COPYRIGHT 1980	APPLE COMPUTER CO.

Copyright 1981, Nestar Systems, Inc.

Then a prompt appears:

```
> (for integer BASIC)
or
] (for Applesoft BASIC)
```

You are now executing DOS BASIC, with one virtual diskette mounted on virtual drive 1 of the slot in which the network card is plugged.

2.5 Issuing Direct File Server Commands Under DOS

Before entering file server commands, it is necessary to issue a command which sends all output to the slot containing the network interface card:

```
PR#n
```

If the slot number is 6, as it usually is, the command is:

```
PR#6
```

Now it is possible to enter Network file server commands and have them executed immediately. The procedure is:

1. Type an @ sign immediately followed by the file server command you wish executed. Press carriage return. (Commands typed without an @ sign are directed to DOS or BASIC just as they would have been without the Network.)

2. a. If the network file server can execute the command, it does so, and responds

```
0,OK.
```

- b. If the network file server cannot execute the command, it returns an error code number and message. For example:

```
1,ILLEGAL COMMAND
30,DRIVE REQUIRED
```

3. The BASIC prompt again appears on the screen.

Note: The numbers which precede the messages are there to be used, for example, when INPUT to applications programs (see Sections 2.7 and 6.4), or as a reference to a more complete description in Appendix A.

In the following sequence, the commands all deal with a virtual diskette named EXAMPLE. If you are following these commands on an Apple, you may prefer to substitute for EXAMPLE the name of a virtual diskette you need to create

MINIMAL SYSTEM

and use.

Now enter:

```
@CREATE /MAIN/EXAMPLE,T=D,SECTORS=16
```

This command will allocate space for a virtual diskette called /MAIN/EXAMPLE. The command also enters the name EXAMPLE and its address in the root directory of MAIN. The field T=D (type= DOS) tells the file-server that this virtual diskette will be used to store DOS programs and data. The field SECTORS=16 tells the file-server to allocate 16 sectors per track for the disk, as required by DOS 3.3. (To create DOS volumes of variable size, see section 6.6.)

(CREATE allows DOS virtual diskettes of any size. However, if less than 16 sectors are present, DOS 3.3 will be unable to INIT the disk; if more than 16 sectors are present, DOS will ignore the excess.)

If the command is entered correctly, the file-server executes it and responds:

```
0,OK
```

You can now verify the CREATE by listing your directory. Enter:

```
@LIST /MAIN
```

The system now lists on the screen all the volumes included in your directory. Among them will be the DOS directory and the EXAMPLE virtual diskette just created:

```
/MAIN
DOS,T=Y
EXAMPLE,T=D
30635S FREE, 8 EXTENTS, 29764S LARGEST
```

The bottom line gives:

the total number of free sectors
on the disk unit

the number of extents--sets of
contiguous free sectors

the number of sectors in the largest
extent of free sectors

After each entry in the simple directory listing, the system

lists the NFS file type:

T=D,P	Virtual Diskette (DOS, Pascal)
T=Y	Directory
T=S	Network File Server File
T=B	Binary File

If your file server has the Nestar Clock/Calendar option installed, the time and date of creation are recorded internally with each virtual file, as well as the times and dates of last modification, access, and backup of the volume. This information is available through the use of the utility programs DOS FILEINFO and Pascal FILEINFO (Chapter 8).

To remove EXAMPLE from your directory, issue a DELETE command:

```
@DELETE /MAIN/EXAMPLE
```

The system should respond:

```
0,OK
```

The volume EXAMPLE is now gone, and you can verify the DELETE by again entering:

```
@LIST /MAIN
```

Now re-enter the CREATE command:

```
@CREATE /MAIN/EXAMPLE,T=D,SECTORS=16
```

and after the

```
0,OK
```

enter a MOUNT command:

```
@MOUNT /MAIN/EXAMPLE,D2
```

To DOS, this is equivalent to inserting a diskette in drive 2 of the slot you booted from. On the Cluster/One system, it involves identifying part of a real network disk so that commands addressed to drive 2 will get and take information from the volume EXAMPLE. As described in Chapter 5, Commands, the CREATE command may also be used to MOUNT new disks at the time of their creation.

After the

```
0,OK
```


MINIMAL SYSTEM

verify the MOUNT by displaying the current MOUNTS at your user-station:

```
@SHOW MOUNTS
```

Now on the screen will appear:

NAME	DRV	ACCESS	T	VOL	SIZE
/MAIN/EXAMPLE	2	RO SHR	D	0	560S
/MAIN/DOS/3.2.1	1	RW UPD	D	254	560S

The system automatically mounted the DOS disk on drive 1 when you began the session on your user-station. It mounted EXAMPLE on drive 2 when you entered the MOUNT command. Because these examples are for a no-frills system, we did not enter any optional parameters for the MOUNT command. Therefore, by default, our virtual disk EXAMPLE was mounted for shared usage (SHR) which means that other users are allowed to mount it at the same time. Also by default, EXAMPLE is mounted with read only access rights. A booted DOS virtual diskette, in this case DOS 3.3, is mounted for read/write (RW), update (UPD) usage, unless the BOOT program has been modified at your installation. (UPD mode allows one user at a time to mount a volume with read/write access, while others can mount it RO.)

To undo the MOUNT (perform the equivalent of removing the diskette from real drive 2) enter:

```
@UNMOUNT D2
```

This can be verified by again entering:

```
@SHOW MOUNTS
```

We have now tried out six fundamental commands:

```
CREATE  
DELETE  
MOUNT  
UNMOUNT  
LIST  
SHOW MOUNTS
```

The next step is to initialize the volume EXAMPLE, so that you can give its name when the system asks VOLUME TO BOOT?

2.6 Initializing a DOS Virtual Diskette

As explained in the "Apple II DOS Manual" (Version 3.3, p.13), a diskette must be initialized before it can be used to store BASIC programs and data. This is also true for virtual diskettes on the Cluster/One Model A. To initialize a virtual diskette, it must be mounted with read/write access.

DOS 3.3 users also have the option of creating and initializing their disks using the Nestar System program VDOS CREATE; see section 6.6.

To initialize EXAMPLE we may mount it (this time on drive 2) with the following command:

```
@MOUNT /MAIN/EXAMPLE,RW,D2
```

The default usage is UPD. Now, following the example on page 14 of the "Apple II DOS Manual" a greeting program, such as:

```
NEW
10 REM GREETING PROGRAM
20 PRINT "VIRTUAL DISKETTE EXAMPLE"
30 PRINT "CREATED today's date BY your name"
40 END
```

Run and debug the greeting program, then initialize the virtual diskette by entering:

```
INIT HELLO,V1,S6,D2
```

This DOS command formats the virtual diskette and saves the greeting program under the name HELLO so that it will be run whenever the virtual diskette is booted.

When the prompt appears again, return to the BOOT program by typing

```
@OFF
```

This time, respond to the prompt

```
VOLUME TO BOOT: /MAIN/
```

with the new pathname:

```
/MAIN/EXAMPLE
```

This time your new volume will be mounted on drive 1 and the screen should display:

VIRTUAL DISKETTE EXAMPLE
 CREATED today's date BY your name

Now you are ready to write, save, and run BASIC and DOS programs, just as you would on a stand-alone Apple. Take a few minutes to verify this with some simple DOS commands. Then write a simple program, save it, display its name in your catalog, and run it.

2.7 Issuing File Server Commands From Within a BASIC Program

All of the file server commands in these examples have been typed directly at the keyboard. It is often convenient to issue the same commands from a program, and this is possible for all of the commands. For example, at the beginning of each session you may want to have the current date displayed on the screen. This can be accomplished by including the file server command

SHOW DATE

in your greeting program. This is also a simple example of how to use file server commands within a BASIC program.

Integer and Applesoft BASIC use slightly different commands to communicate with the network file server. The following example is for Applesoft only. If your Apple has only Integer BASIC, see Section 6.4.2.

Enter the following Applesoft program at your user-station:

```
NEW
10 REM GREETING PROGRAM USING NETWORK COMMAND
20 HOME: VTAB3
30 PRINT CHR$(4); "PR#6"
```

CHR\$(4) is a CTRL-D. It signifies that the next field is a DOS command. "PR#6" enables the network card to look at all subsequent print statements to see whether they contain file server commands.

```
40 PRINT "VIRTUAL DISK EXAMPLE"
50 PRINT "CREATED today's date BY your name"
60 PRINT CHR$(14) "SHOW DATE"
```

CHR\$(14) is a CTRL-N. It signifies that the next field is a file server command. Note that the CTRL-N takes the place of the @ preceding a file server command when it occurs within a program.

```
70 GOSUB 500
```

500 contains a subroutine to input and display the return code and message

```
80 PRINT CHR$(4);"CATALOG"
90 END
```

```
500 REM INPUT AND DISPLAY RETURN CODE AND MESSAGE
510 PRINT CHR$(4); "IN#6"
```

CHR\$(4) again signifies a DOS command. "IN#6" tells the following INPUT command to take input from slot 6 rather than from the keyboard.

```
520 INPUT NN,NN$
```

The file-server sends a return code (NN) and message (NN\$) to the network card in slot 6. These are read by the INPUT statement. After reading the last character from slot 6, the INPUT command resets the input slot so the next input will come from the keyboard.

```
530 PRINT NN;",";NN$
540 RETURN
```

If the network interface card is in some slot other than 6, that slot number must be substituted in the PR#6 and IN#6 commands.

Run and debug the new greeting program. You can replace the previous HELLO program by issuing the DOS command:

```
SAVE HELLO
```

Again reboot by typing
@OFF

This time when the system asks

```
VOLUME TO BOOT?
```

reply with the pathname:

```
/MAIN/EXAMPLE
```


MINIMAL SYSTEM

Your screen should then show:

```
VIRTUAL DISK EXAMPLE
CREATED today's date BY your name

DATE IS 04/01/81

DISK VOLUME 1

A 002 HELLO
A 002 PROGRAM1
```

You now know how to:

```
CREATE a virtual diskette

MOUNT it on a virtual drive

Initialize it

Boot it

Issue commands and receive responses from
within programs
```

This means you are ready to use your Apple computer just as you do a stand-alone Apple. You can write programs and save them on your virtual diskette, then LOAD and RUN them, just as you have always done, but now using shared disks in a local network of Apples.

2.8 Copying a Program Library

If you already have a library of programs and data files for your Apple, either on cassette tapes or diskettes, you can copy them onto your virtual diskette just as you would with a stand-alone Apple.

2.8.1 Copying From a Cassette Tape

1. Create and mount the virtual diskette to which you wish to copy files. (We assume the disk will be mounted with read/write access on drive 1 and the network interface card is in slot 6.) For example:

```
MOUNT /MAIN/EXAMPLE,D1,RW
```

2. Connect the tape cassette to the special plug at the rear right of the Apple.

3. Give the DOS commands:

LOAD

SAVE filename

(See the DOS manual for more details on loading programs from a cassette.) Repeat the two DOS commands in step 3 until all your files are copied.

2.8.2 Copying from a Diskette

NOTE: Chapter 8 describes the Nestar Disk Copy program, which copies whole disks (including data files). The following procedure is to be used when only selected programs are to be copied.

1. Create and mount the virtual diskette to which you wish to copy files. Be sure the number of sectors per track is the same for the real and virtual diskettes--i.e., they MUST both have been initialized under DOS 3.3 (16 sectors per track) or under an earlier version of DOS (13 sectors per track). (We assume that the network card is in slot 6 and that the virtual diskette is mounted with read/write access on drive 1.)

2. Connect the diskette to either drive of a disk control card inserted in any available slot. Here we use drive 1, slot 5.

3. Give the two DOS commands:

LOAD filename,S5,D1 (from real diskette)

SAVE filename,S6,D1 (to virtual diskette)

Repeat the DOS commands in step 3 until all the files are copied.

2.9 Other Features

For many applications, what has been described so far is all you need to know. But the Network File Server also serves the user who needs to:

automatically mount virtual diskettes at Power On (Chapter 8)

organize available space in the network's shared memory into directories and virtual diskettes of various types and sizes (Chapter 3)

protect directories and NFS files by requiring

MINIMAL SYSTEM

passwords and defining access rights for different classes of users (Chapter 4)

allow several users to set locks in order to cooperate in sharing system resources (Chapter 4)

use binary files (Chapter 5)

use the Pascal Operating System (Chapter 7)

These features are described in the remainder of this manual.

Chapter 3

Disk Organization and Pathnaming Conventions3.1 Disk Organization3.1.1 Disk Units

The file server manages disk space which consists of some combination of Nestar 8-inch floppy and hard disk units (Product Models A-2001, A-2002, and A-2003). Each disk unit is referred to by either its assigned name or its unit number. Disk units may be given names when they are formatted by a Nestar-supplied utility program. Unit numbers are fixed as follows:

Unit #	Unit or Device
1	disk in left drive of floppy disk (removable diskette)
2	disk in right drive of floppy disk (removable diskette)
5	hard disk (nonremovable volume)
6	hard disk (nonremovable volume)

3.1.2 Files

The file structure on each physical disk unit has two distinct levels:

1. the Network level, a hierarchical collection of directories and files. (This level is unknown to DOS and Pascal, and is equivalent to a set of cardboard boxes, each containing a number of real diskettes.)
2. the DOS and Pascal level, where program and data files are stored within a single (virtual) diskette.

The Network-level files on each disk unit consist of:

one root directory

directories

virtual diskettes

binary utility programs and system data files

Note: In this manual, the expression "NFS file" will refer to these four types of Network-level files. The word "file" alone will be reserved for the program and data files contained within a single virtual diskette and controlled by

DISK

the DOS or Pascal operating systems. The expression "virtual diskette" is equivalent to "volume" as used by DOS and Pascal. The expressions "File Server", and "file-server" are used interchangeably to refer to an Apple that executes file server commands and transfers information to and from shared disks.

3.1.3 Directories

The root directory on a given physical disk unit points, either directly or through directories, to every virtual diskette, utility program, or directory on that unit. It cannot point to NFS files on other physical units.

Each root directory and directory consists of names and disk addresses of directories, virtual diskettes and/or Network utility programs on the same physical unit as the directory itself. A directory which contains the name and address of a given NFS file will be referred to as the parent directory of that NFS file.

Directories are automatically expanded as necessary.

3.1.4 Virtual Diskettes

Virtual diskettes are of two types:

DOS
Pascal

Virtual diskettes can vary in size, as specified by their creator. See Chapters 6 (DOS) and 7 (Pascal) for details.

3.1.5 Network Utilities

File Server utility programs are used to:

start (boot) the file server
start (boot) a user station
copy and format system floppies and hard disks
back-up the system
set the clock/calendar
provide information about existing NFS files
create virtual DOS disks of variable size

Most utilities reside as files within Pascal virtual volumes on the Nestar disks distributed with the Model A system; some, like /MAIN/BOOT, are NFS binary files. User-station utilities are described in Chapter 8. File server utility programs are described in the System Manager's Manual.

3.2 Pathnaming Conventions

To address an NFS file on the Model A system, it is necessary to specify its "pathname" by some combination of defaults and explicit names. These consist of:

the physical disk unit (i.e its root directory),
either by name or by number

intermediate directories (if any)

the name of the desired NFS file

To address the root directory itself, only the physical disk unit number or name needs to be specified.

Here are a few examples of how pathnames are used with the LIST and CREATE file server commands:

3.2.1 Pathname: /unitnumber

LIST /1

This command lists all the NFS files whose names and addresses are included in the root directory on disk unit 1 (the 8-inch diskette inserted in the left-hand drive of the first system floppy disk unit). The initial slash indicates that the first field is the name or number of a physical disk unit.

3.2.2 Pathname: /unitname

LIST /MAIN

This command lists all the NFS files whose names and addresses are included in the root directory of the system floppy disk or hard disk called MAIN (regardless of which physical disk unit it is located on).

3.2.3 Pathname: /unitname/directory

LIST /LANGUAGES/FRENCH

This command searches for the physical disk unit named LANGUAGES, and looks in its root directory for the name and address of the directory FRENCH. It then reads FRENCH and displays all the files listed in it.

DISK

3.2.4 Pathname: /unitname/directory1/directory2/virtual diskette

```
CREATE /MAIN/USERS/JANE.DIR/JANE.V1,T=P
```

This command:

searches for the physical disk unit MAIN

looks in its root directory for the directory
USERS

looks in USERS for JANE.DIR

allocates space on the physical disk unit MAIN
for the virtual diskette JANE.V1

enters the name and address of JANE.V1 in the
directory JANE.DIR.

If the directories USERS and JANE.DIR in the above example
do not exist, the CREATE command creates them as needed.

3.2.5 Current Default Directory

In order to cut down on the number of disk accesses and the
number of characters to be typed and processed, the user may
specify a default directory. Whenever a pathname is
specified which does not begin with a slash, it is prefixed
by the current default directory. In the previous example,
assume that the directories USERS and JANE.DIR exist before
the CREATE command is issued. If so, Jane will achieve the
same result if she enters:

```
SET DIR /MAIN/USERS/JANE.DIR  
CREATE JANE.V1,T=P
```

The SET DIR command above sets the current default directory
to:

```
/MAIN/USERS/JANE.DIR
```

which is prefixed to the JANE.V1 when the CREATE command is
executed. Thus the create command above is equivalent to:

```
CREATE /MAIN/USERS/JANE.DIR/JANE.V1,T=P
```

The SET DIR, LIST, PROTECT, and SHOW PROTECTION commands can
be specified with no pathname; if so, each operates on the
current default directory. A SET DIR with no pathname
clears the current default directory. The SHOW DIR command
displays the name of the current default directory.

3.2.6 Passwords

As will be explained more fully in Chapter 4 (Protection), it is possible when creating an NFS file to require that a password be specified each time the NFS file is accessed for some or all operations. In our last example, let's suppose the passwords needed are:

```
COFFEE (for the root directory MAIN)
TEA    (for the directory USERS)
MILK   (for the directory JANE.DIR)
```

If her current default directory is set for some other directory, Jane would enter:

```
CREATE /MAIN:COFFEE/USERS:TEA/JANE.DIR:MILK/JANE.V1,T=P
```

Or alternatively, she can enter:

```
SET DIR /MAIN:COFFEE/USERS:TEA/JANE.DIR:MILK
CREATE JANE.V1,T=P
```

It is also possible to set a password for the disk JANE.V1, and to set default passwords so that they do not have to be entered each time the pathname is specified; these options are explained in Chapter 4.

3.2.7 Pathnames--A Formal Definition

In general terms, a pathname is specified according to the following definitions. Square brackets indicate optional parameters, e.g. [:password].

```
pathname: /unitname/filepath
          or filepath
```

```
unitname: 1, 2, 5, or 6 [:password]
          or name [:password]
```

```
filepath: name [:password] [/filepath]
```

```
name:     the name of a disk unit (root
          directory), directory, virtual
          disk or Network utility. A name
          has the following restrictions:
```

1-15 characters.

No commas, slashes, colons,
carriage returns, ASCII control
characters, or unprintable

DISK

characters. Acceptable characters: 0-9, A-Z, !, ", #, \$, %, &, ', (,), *, <, >, ?, [,], {, }, ^, ~, @, _, |

Upper and lower case letters are not distinguished (ABC=abc=Abc=)

Imbedded blanks are okay.
Leading and trailing blanks are removed automatically.

The first character can be any acceptable character.

password: a word used to restrict access to a directory or virtual file. Same restrictions as name.

Note that this is a "recursive" definition in that a form for filepath includes a filepath. Another way to describe a legal filepath is to say that it consists of one or more names separated by slashes, and each name may optionally be followed by a colon and a password. Recursive definitions will be used elsewhere in the manual to concisely describe sequences of terms.

Four other rules apply to the pathname:

1. The total pathname cannot exceed 80 characters (including /'s).
2. Upper or lowercase can be used for any characters within a file server command, and are equivalent.
3. An initial slash indicates the pathname begins in the root directory of a physical disk unit. If there is no initial slash, the current default directory is prefixed to the pathname.
4. Blanks before and after the slashes in a pathname are optional and are ignored. Blanks within file or directory name are part of the name.

Chapter 4

Protection

The Cluster/One Model A Network File Server allows many users to share a large data base, something that is barely feasible for stand-alone Apples. In a classroom, students can share a large selection of courses; in a factory, many different user-stations can update a central inventory; in a business office, the billing and advertising departments can share an address list. However, universal access is not always desirable: files must not be accessed at sensitive times or by unauthorized users. A payroll application must foil the potential computer crook; a classroom manager must make sure the students don't change their grades or copy the answers to a final exam; and inventory programs must make sure two users are not updating the inventory at the same time.

The Network File Server fully controls user-station access to the shared data base and manipulation of the data. User stations cannot breach the security provided by Network protection by employing, for example, PEEKs and POKEs.

The Network File Server has three different software mechanisms for protecting the shared resources of the system:

1. NFS file protection, a system of passwords and access rights for different classes of users
2. Usage modes, a means of temporarily reserving file system resources for exclusive or restricted use.
3. Locks, which allow users to synchronize access to any resource; for example, external devices or individual files on a virtual diskette.

These three mechanisms differ in the following ways:

Passwords and access rights control the usage of NFS files, including virtual diskettes and directories. The protection is involuntary: passwords and access rights allow only the usage that was specified for the virtual diskettes and directories when the protection was set.

Usage modes allow the user to temporarily restrict access to given virtual diskettes and directories (that is, to restrict access while the disks and directories are

PROTECTION

mounted). Volumes can be mounted for:

- (a) exclusive read/write (EXC)
- (b) exclusive write and shared read (UPD)
- (c) shared read/write (SHR)

Locks can be used to control access to any resource--e.g., a printer, plotter, scientific instrument, or individual file on a virtual diskette. Locks are voluntary; they provide a convention that cooperating stations can use. The system returns the information that a lock is in use or in exclusive use, but it does not prevent the user from addressing the printer, the particular file, or whatever he or another user is trying to protect.

4.1 NFS File Protection

Each NFS file can have three different classes of users who have different passwords and are allowed to do different things. When an NFS file is created, the password required and the access rights allowed for each class of user are specified.

This section describes:

the user classes and access rights

how to assign passwords and access rights with the CREATE and PROTECT commands

what the default access rights are for each class of user

preset passwords

access rights needed to execute various file server commands

4.1.1 User Classes

Potential users of a given NFS file are divided into three classes:

- | | | |
|---------|-------|--|
| public | (PUB) | users who do not specify a password |
| group | (GRP) | users who specify the group password |
| private | (PRV) | users who specify the private password |

The access rights which may be granted or denied to each

class of user are:

R the ability to read the data part of the NFS file

W the ability to write on the NFS file

E the ability to erase (remove completely) the NFS file

C the ability to create a new entry in the directory
(meaningful only for directories)

D the ability to delete an entry from the directory
(meaningful only for directories)

An NFS file is removed from the system by the DELETE command. To accomplish this two operations are done: the file itself is removed (erased), and the entry in the directory which pointed to the file is removed (deleted). The E (erase) access right is required to erase the file; the D (delete) access right is required in the corresponding directory to delete the directory entry.

For example, suppose a given Cluster/One installation wants to have accounting programs on a Pascal volume called ACCOUNTING. They want ACCOUNTING to be listed in a directory called ADMINISTRATION, which in turn will be listed in the root directory of the system hard disk.

They want to limit access to the directory and volume as follows:

1. public: the public will have no access rights.

2. group a select group of employees will be able to:

R read the virtual diskette and
execute the programs on it

The group password will be SILVER.

3. private The person in charge of the programs will
be allowed to:

R read the virtual diskette

W write on the virtual diskette

E erase (remove completely) the virtual
diskette

He will also be allowed to:

PROTECTION

D delete entries from the directory

C create new entries in the directory

The private password will be GOLD.

The access and usage rights for this example are established in the following section.

4.1.2 Passwords and Access Rights

Passwords and access rights can be specified for a new NFS file by a CREATE command and for an existing NFS file by a PROTECT command. With [optional] parameters indicated by square brackets, the syntax of these two commands is:

CREATE pathname, type [,size] [,protection] [,drive] [,usage]

PROTECT pathname [,protection]

For a complete discussion of CREATE and PROTECT see the individual command descriptions in Chapter 5. For an explanation of the [,usage] option, when CREATE is used to mount a new disk, see 4.2.

The optional protection parameter is the same for both CREATE and PROTECT and is defined as follows:

protection: protect item [,protection]

protect items:

GRPPW=password (group password)

PRVPW=password (private password)

PUBACC=access rights (public access rights)

GRPACC=access rights (group access rights)

PRVACC=access rights (private access rights)

access rights: a subset (in any order) of the letters RWECD (read, write, erase, create, and delete, as described under "private" in above example)

To specify no access rights, the equal sign is immediately followed by the delimiter--comma, blank or carriage return.

For example:

```
PUBACC=,GRPACC=
```

For the above example, the CREATE command would be:

```
CREATE /MAIN/ADMINISTRATION/ACCOUNTING,T=P,PRVPW=GOLD,  
PRVACC=RWE,GRPPW=SILVER,GRPACC=R,PUBACC=
```

This command creates a Pascal volume named ACCOUNTING which is listed in the subdirectory ADMINISTRATION which in turn is listed in the root directory of the hard disk. Users giving the password GOLD can read, write or erase ACCOUNTING, but users giving the password SILVER can only read it. Users who do not give the password GOLD or SILVER have no access rights. Users give the password in the form ACCOUNTING:GOLD or ACCOUNTING:SILVER when specifying the pathname in their file server commands. Users must give the password ADMINISTRATION:SILVER or ADMINISTRATION:GOLD when specifying the directory. (Pathnames are discussed in detail in Chapter 3.)

The CREATE command shown above also implicitly creates the directory ADMINISTRATION (assuming it does not already exist). Passwords and access rights for ADMINISTRATION are the same as for ACCOUNTING, except that create and delete access rights are automatically added to the private rights for the directory. Note that passwords must be given, when set, for each directory and file in the pathname.

For instance, the volume in this example could be mounted for writing by

```
MOUNT /MAIN/ADMINISTRATION:GOLD/ACCOUNTING:GOLD,RW,D9
```

4.1.3 Default Access Rights

If public, group, or private access rights are not specified in a CREATE command, the Cluster/One system sets standard default values:

```
PUBACC=R (read)
```

```
GRPACC=R (read)
```

```
PRVACC varies:
```

For a virtual diskette: RWE (read, write, erase)

For an explicitly created directory: RWECD (read, write, erase the directory, create and delete

PROTECTION

entries in the directory)

For an implicitly created directory: the private access rights for the NFS file being explicitly created plus CD--create and delete.

Therefore, in the previous example, the field defining private access rights is not necessary because they are the same as the defaults.

Warning: If an NFS file has no private password, then all of its private access rights are granted to any user, whether he gives no password or some extraneous password. This means that if a NFS file is created with group password and access rights only, public users will have, by default, private access rights: read, write, and erase. In other words, for group passwords and access rights to have meaning, private passwords and access rights must also be set.

When an NFS file is created, all directories specified in the pathname which do not already exist are also created; these implicitly created directories are given CD (create and delete) private access rights plus the access rights specified for the NFS file being explicitly created; they are also given the same private passwords.

4.1.4 The PROTECT Command

Usually access rights and passwords are set when an NFS file or directory is first CREATED. The PROTECT command allows the user to modify the access rights and passwords previously set. The syntax is given in Section 4.1.2 and Chapter 5. Protect items which are not specified by the PROTECT command are not changed. Anyone giving the private password for the specified NFS file can issue a PROTECT command; access rights do not apply. If there is no private password then anyone can issue the PROTECT command.

4.1.5 The SHOW PROTECTION Command

The SHOW PROTECTION command displays the private, group and public access rights currently assigned to the specified NFS file. Passwords are never displayed, but their existence is indicated by ,PW. The access rights for the current default directory may be displayed by entering SHOW PROTECTION with no pathname.

4.1.6 Preset Passwords

The SET command allows a user to preset the group and private passwords to be used in subsequent access commands, so that it is not necessary to enter them each time a pathname is specified:

```
SET GRPPW=x
SET PRVPW=y
or SET GRPPW=x,PRVPW=y where x and y are passwords
```

This command makes it less likely that an unauthorized person will learn a password and will also cut down on the amount of typing to be done when entering a pathname.

If passwords are in a program, others can read the listing and know what they are. For example, if a password is WOW and your program contains the two file server commands:

```
MOUNT /MAIN/GRADES:WOW,D9"
MOUNT /MAIN/ANSWERS:WOW,D10"
```

an enterprising student can read your program and learn the password.

Instead, the following Applesoft BASIC program segment shows how passwords can be used in a program without appearing in the listing:

```
PRINT "ENTER PASSWORD";
INPUT PW$
PRINT CHR$(14);"SET PRVPW=";PW$
PRINT CHR$(14);"MOUNT /MAIN/GRADES,D9"
PRINT CHR$(14);"MOUNT /MAIN/ANSWERS,D10"
```

4.1.7 Access Rights Needed for Network Commands

In order to execute a file server command with a pathname parameter, a user must have the READ access right for every directory in the pathname (Except for CREATE, where the lowest directory may have only CREATE access). He needs additional access rights for the following commands:

<u>Command</u>	<u>Access Rights Required</u>
CREATE	C (create) for parent directory
DELETE	E (erase) for NFS file or directory being erased D (delete) for parent directory
RENAME	C (create) and D (delete) for parent

PROTECTION

	directory
PROTECT	Must have private password for NFS file. Access rights irrelevant.
MOUNT	R (read) for directories and other NFS files in pathname W (write) for NFS file to be mounted, if RW specified.
BRUN	
BLOAD	R (read) for NFS files addressed by pathname
LIST	
SHOW PROTECTION	
SHOW DIR	
SHOW TYPE	
BSAVE	W (write) for virtual diskette addressed by pathname

NFS files may be created in a directory (added to that directory) if you have C(reate) access rights, even if you don't have R(ead) or W(rite) access rights. You will not be able to MOUNT such files, however, so you should use the form of the CREATE command that creates and mounts the file in one operation if you intend to write into it. Once the file is unmounted, you no longer have access to it. You cannot LIST a directory to which you don't have R(ead) access.

Note that the DELETE command requires two different access rights for two different objects: E (erase) rights for the object to be removed, and D (delete) rights for the parent directory of that object. As an example, suppose that the directory FOO lists three virtual volumes: A, B, and C. To DELETE A, B, or C, the user must have not only the right to erase (remove completely) A, B, or C, but also the right to delete A, B, C from the directory FOO. DELETE, then, erases (removes completely) a virtual volume, directory, or other NFS file and deletes the directory entry that pointed to it. To DELETE (remove) the directory FOO, the user would need erase rights for FOO and delete rights for its parent directory.

4.1.8 What Happens If You Forget a Password...

Passwords are actually stored in an encrypted form from which it is impossible to reconstruct the original name. If you forget a password, there is no way to discover it by dumping the disk. The only recourse is to have the System Manager use superuser rights to remove or change the

password.

4.2 Usage Modes and the MOUNT Command

In addition to the restricted rights which are controlled by the protection attributes of an NFS file, access rights may be (voluntarily) limited even further when the virtual diskette is actually used. The MOUNT command (or the CREATE command, for newly created disks) allows a user to mount a specified virtual diskette with either read-only or read-write access, provided that these are access rights his password entitles him to. For example, if a user gives the private password and has full access rights, he may want to mount his virtual diskette with read only rights so as to prevent accidental writes. Usage is an optional parameter for the MOUNT and CREATE commands:

[,usage]

where usage can be RO (read only) or RW (read/write) and SHR (shared), EXC (exclusive), or UPD (update).

UPD is an Exclusive-Write/Shared-Read mode. Thus, only one UPD mount of a file (either RW or RO) is allowed at a time, but other users may mount the same file RO, SHR.

If incomplete access rights are given when mounting a volume, the following defaults are used:

Usage Specified	Default Applied
RO	SHR
RW	UPD
SHR	RO
UPD	RW
EXC	RO
none	RO,SHR

4.3 Locks

LOCK commands allow several programs to cooperate in sharing a resource on the Cluster/One system, such as a particular file, printer, or some other external device. Sharing a printer is an obvious application, since it would be less than useful for output from two programs to be printed at the same time.

Another example is using the Cluster/One for inventory control. If two users update the inventory at the same time, one of the transactions may be lost. Suppose a computer store has eight Apples on hand; employee A sells two Apples; employee B sells 3 Apples; and both employees enter their transactions at the same time.

PROTECTION

For the first transaction, the computer reads the file, computes $8-2=6$, and sends the result (6) back to the file. For the second transaction, the computer reads the file before the 8 is changed to a 6, computes $8-3=5$, and sends its result (5) to the file. So the computer now erroneously thinks it has 5 instead of 3 Apples on hand.

4.3.1 LOCK Command

To avoid such errors, each program, before updating the inventory, can issue a lock command:

```
LOCK lockname [,usage]
```

where lockname is any name 1-15 characters in length (except ALL) which has been chosen to represent the resource being shared. Usage is either:

```
SHR  shared--more than one user can have
      the lock set at a given time
or
EXC  exclusive--only one user can have
      the lock set at a given time.
```

The default for usage is EXC.

The system then returns one of the following messages:

```
0,OK
```

```
41,IN USE
```

```
42,IN EXC USE
```

```
(or some syntax error messages)
```

The IN USE message is returned if the lock command specifies EXC (exclusive) usage when another user has already set the specified lock. The IN EXC USE message is returned if the lock command specifies SHR (shared) usage when another user has set the specified lock for exclusive use.

If the program receives the 0,OK message it can proceed to use the printer or update the file or whatever, but if it receives a 41,IN USE or 42,IN EXC USE message it should either go into a wait loop or go on to some other task before reissuing the LOCK command.

4.3.2 UNLOCK Command

To release a lock, one of the following commands is issued:

UNLOCK lockname

-releases lockname for use by LOCK
commands from other stations

or

UNLOCK ALL

-releases all locks held by this station

The locks held by a station are automatically unlocked when the station is booted. In particular, turning the station off and on or issuing the OFF command will release all locks.

4.3.3 SHOW LOCK(S) Command

A user can display locks in several ways:

SHOW LOCKS

-displays all locks currently set at
his own user station

SHOW LOCK lockname

-displays the station numbers of all
user stations currently using the
specified lock

SHOW ALL LOCKS

-displays all locks currently being
used, and by which stations

Chapter 5

Network File Server Commands

A Network file server command consists of a command word followed by a variable number of parameters, some required and others optional. A pathname or subcommand word, if present, must immediately follow the command word; other parameters may occur in any order. The delimiter following a command word must be a blank and the delimiter following a pathname must be a comma; elsewhere, a comma is used as delimiter.

Any alphabetic character in a file server command can be either upper or lower case; they have the same meaning. For standard Apples, this is not useful, but for Apples with terminals which display upper and lower case, and from Pascal, it is convenient.

Remember that file server commands may have to be preceded by special characters, depending on the language environment. Briefly, these characters are:

@	for the BASIC command level
CHR\$(14)	for the BASIC program level
nothing	for the Pascal CMD program
nothing	for the Pascal SENDMSG function

See Chapter 6 (DOS) or 7 (Pascal) for further details.

Syntactically, a file server command has the following form:

```
<0-n blanks><command word><1-n blanks><subcommand word>
<1-n blanks><pathname><,other fields>
```

A pathname may have imbedded blanks, so a comma must be used to delimit it from the following fields. No blanks may be imbedded in a

command word
subcommand word
other fields

See Section 3.2 for a full description of pathnames.

COMMANDS

A field may have one of three forms:

Keyword (e.g. REPLACE or ALL)

Keyword=Value (e.g. T=D)

Letter value (e.g. D9 for drive 9)

Optional parameters are indicated by square brackets, e.g. [optional]. The system assigns default values, where meaningful, when optional parameters are not specified; these are given in the descriptions of the individual commands.

Some of the parameters are explained more fully elsewhere in this manual:

pathnames	Section 3.2
protection	Section 4
size	
DOS	Section 6.1
Pascal	Section 7.1
drive number	
DOS	Section 6.2
Pascal	Section 7.2

In addition, a one-page summary of PATHNAMES and PROTECTION are filed alphabetically with the commands in this chapter.

A quick reference list of all available file server commands and parameters occurs at the end of this book. The commands and parameters are alphabetized, and a page reference to detailed descriptions follows each item. A separate fold-out reference card with a summary of all commands and options is available from Nestar (Publication number #GX20-0101-0).

File server commands can be divided into these classes:

1. Commands concerned with creating, deleting, naming and protecting virtual NFS files (virtual volumes):

CREATE
DELETE
LIST
RENAME
PROTECT
SHOW PROTECTION

2. Commands concerned with mounting virtual diskettes at a user station:

a. For both DOS and Pascal

MOUNT
UNMOUNT
SHOW MOUNTS

b. For Pascal only. Executed by CMD (see Section 7.3) or within a user's own program (Section 7.4):

SHOW DRIVES
SHOW SLOT
SET S<slot>
SET D<drive>,S<slot>
SET D<drive>,drive-status

3. Commands concerned with binary files

BLOAD
BRUN
BSAVE

4. Commands concerned with locks

LOCK
UNLOCK
SHOW LOCK lockname
SHOW LOCKS

5. Commands which set default values

SET DIR
SET GRPPW
SET PRVPW

6. Commands which gives information at terminal

HELP
?
SHOW DATE
SHOW DIR
SHOW STATION
SHOW TYPE(S)
SHOW VOLS
TIMESTAMP

COMMANDS

The remainder of this chapter consists of detailed descriptions of each file server command. The commands are presented in alphabetical order, and each new command or group of commands starts on a new page. For each command, the following information is presented:

the command syntax

an abbreviated statement of what the command does

parameters and defaults

error and information messages

access rights required to execute the command

discussion

examples

All error and information messages are listed in Appendix A. Individual command descriptions include only those messages which are specific to the given command.

BSAVE pathname, FROM=number, size, [,LOAD=number] [,REPLACE]

- saves a memory image of specified size
starting at FROM address in memory,
to the specified binary virtual disk

BLOAD pathname [,LOAD=number]

- loads binary virtual disk previously saved
by BSAVE
- loads at the LOAD memory address in BLOAD, if
given; if not, loads at LOAD address in
corresponding BSAVE, if given; otherwise,
loads at FROM address in corresponding BSAVE

BRUN pathname [,LOAD=number]

- BLOADS a binary virtual disk, then executes it
starting at first address loaded

Parameters and defaults:

pathname: For summary, see PATHNAME, filed alphabetically
with commands in this chapter. For discussion,
see Section 3.2

[LOAD=number]

number: if decimal: between -32768 and 32767, inclusive
if hex: between \$0 and \$FFFF, inclusive

address at which memory image is to be loaded

LOAD address used for BLOAD and BRUN, in order of priority:

- LOAD address given by BLOAD or BRUN
- LOAD address given by corresponding BSAVE
- FROM address given by corresponding BSAVE

FROM=number

starting address of memory image being saved

BLOAD, BRUN, BSAVE

SIZE=number [.size-unit]

For BSAVE the amount of memory to be saved must be specified with a SIZE parameter. The size given must be less than or equal to the size that was given in the CREATE of the binary file.

Size may be given in units of bytes (characters), sectors (256 bytes each), blocks (512 bytes each), or K (1024 bytes each). The default unit for size is blocks. The size parameter has the form: SIZE=number[.unit]. Unit may be 'C' for characters, 'S' for sectors, 'B' for blocks, or 'K' for K. The '.' is optional except where it is needed to resolve ambiguity in the specification of hexadecimal numbers. It is suggested that the '.' be used for readability.

[,REPLACE]

If a memory image has previously been BSAVED using the same pathname, it will not be destroyed and the present BSAVE will not be executed unless the REPLACE parameter is present. (REP is also recognized.)

Error Messages

50,CURRENTLY MOUNTED
51,NOT BINARY
52,NOT PREVIOUSLY SAVED
53,FROM PARAMETER REQUIRED
54,SIZE PARAMETER REQUIRED
55,FILE WAS THERE REP NOT SPECIFIED

Access Rights Required

R (READ) for each filename in pathname
W (WRITE) for file being BSAVED

Discussion

These three file server commands (BLOAD, BRUN, and BSAVE) allow machine language programs to run independently of DOS and Pascal. They are especially useful when customizing a BOOT program (which runs before DOS or Pascal is read in) or for running programs without making assumptions about the currently loaded operating system, if any.

It is also possible, just as with a stand-alone Apple, to run machine language programs under DOS and Pascal using the commands already in those language systems.

Before a BSAVE is issued, a binary-type virtual disk of sufficient size must be CREATED. Then a BSAVE is issued which stores a memory image on the virtual disk. The user specifies the starting address (FROM=number) of the memory image and the size of the binary Network-file to be saved. An optional parameter, [,LOAD=number], allows the user to save a memory image from one part of memory and load it (with a BLOAD or BRUN) into a different part.

BLOAD and BRUN load a memory image previously saved by the BSAVE command. The first address loaded is the LOAD address of the BLOAD or BRUN command, if specified. Otherwise, it is the LOAD address of the BSAVE command, if specified, or the FROM address of the BSAVE, if neither LOAD address is specified.

BRUN does a BLOAD, then executes the loaded program starting at the first address loaded.

Note that it is not necessary to MOUNT a binary virtual disk before issuing the BSAVE, BLOAD or BRUN command.

The NFS "B" commands differ from the DOS commands of the same name in that (1) DOS need not be present in the machine when the commands are used; (2) the file being loaded, saved, or run is not one of several files within a DOS virtual disk. It is a virtual disk itself, defined only by its NFS pathname.

Examples

1. Data Collection

Suppose a meteorologist uses his Apple to measure cloud patterns on satellite weather pictures. He inputs data from a picture reading device with a machine language program. To do this, he:

- a) creates a binary virtual disk of the correct size

```
CREATE READIMAGE,T=B,SIZE=20.S
```

- b) enters and debugs his machine language code

- c) BSAVES the code on the previously created virtual disk

```
BSAVE READIMAGE,FROM=$800,SIZE=4096.C
```

- d) executes the program with a BRUN

```
BRUN READIMAGE
```

BLOAD, BRUN, BSAVE

2. Editing in different part of memory

If the meteorologist wishes to edit his machine language program, it is sometimes convenient to read it into a part of memory different from where it will reside when executed. In the above example, he might:

- a) read the program into a different part of memory

BLOAD READIMAGE,LOAD=\$2800

- b) make desired changes in the program

- c) again save the program on disk

BSAVE READIMAGE,FROM=\$2800,SIZE=4096.C,LOAD=\$800

- d) again run the program

BRUN READIMAGE

or, alternatively:

- c) BSAVE READIMAGE,FROM=\$2800,SIZE=4096.C

- d) BRUN READIMAGE,LOAD=\$800

CREATE pathname,type [,size] [,sectors] [,protection] [,drive] [,usage]

-creates (and mounts, if a drive parameter is given) a new file with specified type, size, drive, passwords, access rights, and usage rights

Parameters and defaults:

pathname: For summary, see PATHNAME filed alphabetically with commands in this chapter. For discussion, see Section 3.2

type:

T=B (binary)
T=D (DOS)
T=P (Pascal)
T=Y (directory)

[,size]

SIZE=number [.size-unit]

Size of the file to be created may be given in units of bytes, sectors (256 bytes each), blocks (512 bytes each), or K (1024 bytes each). The default unit for size is blocks. The size parameter has the form: SIZE=number[.unit]. Unit may be 'C' for bytes, 'S' for sectors, 'B' for blocks, or 'K' for K. The '.' is optional except where it is needed to resolve ambiguity in the specification of hexadecimal numbers. It is suggested that the '.' be used for readability.

CREATE allows variable sized files to be created within constraints imposed by the environment in which they will be used (DOS, Pascal, etc.) The following table shows the algorithm used to determine the size of a file being created given the SIZE=, T=, and SECTORS= parameters specified.

<u>T=</u>	<u>SECTORS=</u>	<u>SIZE=</u>	<u>default</u>	<u>limits on SIZE=</u>
Y	ignored	allowed	3.s	0 < SIZE < 32767.S
B	ignored	required	-	0 < SIZE < 193.S
P	ignored	allowed	560.s	0 < SIZE < 64535.S
D	required	allowed	if SECTORS=13 then 455.S else 560.S	0 < SIZE < 32767.S

CREATE

<u>T=</u>	<u>Resulting Size</u>
Y	SIZE
B	SIZE
P	SIZE rounded up to multiple of 2 sectors
D	SIZE rounded up to even # of tracks, each with SECTORS sectors

Note that the actual disk allocation may be slightly higher than the sizes listed in the table. For virtual disk type volumes, one extra sector is added by the file server for file descriptor information.

For directories (T=Y), each sector is big enough for ten names, so the default size of three sectors will hold thirty names. Directories will be automatically expanded when required, so that it is usually not necessary to initially allocate directories more than three sectors.

For a further discussion of size, see Sections 6.1 and 6.6 (DOS and VDOS CREATE) and 7.1 (Pascal).

[,sectors]

SECTORS=13 required for DOS 3.2.1
SECTORS=16 required for DOS 3.3, unless VDOS CREATE is used
 (see Section 6.6)

The number of sectors is ignored for other than DOS-type virtual disks.

[,protection] See Chapter 4.

protection: protect item [,protection]

protect item: ,GRPPW=password
 ,PRVPW=password
 ,PUBACC=access rights (Default: R)
 ,GRPACC=access rights (Default: PUBACC)
 ,PRVACC=access rights (Default: RWE, virtual
 diskette; RWECD, explicitly created direct-
 ory; private access rights of created NFS
 file, plus CD, implicitly created directory)

access rights: set of letters from RWECD (read, write,

erase, create, and delete)

[,drive]

Dd, where d=number (1-255 for DOS; 3-12 for Pascal 1.1)

[,usage]

Specify Type of Access:

RO (read only)

RW (read/write--must have write access)

Default: RO (RW for UPD)

Specify User Class:

EXC (exclusive)

SHR (shared)

UPD (RW usage for only one user at a time; other users may mount the volume RO,SHR)

Default: SHR, if RO or no usage parameter given
UPD, if RW

Directories (T=Y) and system files (T=S)
may be mounted for read only (RO) access.

See also MOUNT command.

Error Messages

13,ILLEGAL NUMBER (NEGATIVE OR TOO BIG)

20,TYPE PARAMETER REQUIRED, (T=)

21,SIZE OUT OF RANGE

23,SIZE PARAMETER REQUIRED WITH T=B

24,SIZE OUT OF RANGE (<1C OR >48K)

25,SECTORS PARAMETER REQUIRED WITH T=D

Access Rights Required to Execute a CREATE

C (create) for parent directory

R (read) for each directory on pathname except
the parent directory

NFS file names may be created in (added to) a directory if you have C(reate) access rights, even if you don't have R(ead) or W(rite) access rights. You will not be able to MOUNT such files, however, so you should use the form of the CREATE command which creates and mounts the file in one operation if you intend to write into it. Once the file is unmounted, you no longer have access to it.

CREATE

Discussion:

The CREATE command:

1. allocates space for specified virtual disk or directory on specified (or default) physical disk unit
2. creates intermediate directories as necessary
3. enters its name in directory preceding it in pathname, or, if not specified, in current default directory
4. stores passwords and access rights on virtual disk or directory being created; these are stored in a special section not accessible to the ordinary user
5. optionally mounts the newly created disk for subsequent I/O

For directories implicitly created because they do not already exist, users are given CREATE and DELETE access rights plus the access rights and passwords specified for the file being explicitly created.

Virtual disks are stamped with time and date upon creation. FILEINFO (section 8.1.3) and DOS FILEINFO are used to display this information.

Examples

1. CREATE /1/USERS/NICK/NICK.V1,T=D,
PRVPW=GOODSTUFF,GRPACC=,PUBACC=,SECTORS=13

This example creates a virtual disk called NICK.V1 on disk unit 1, the disk inserted in the left-hand drive of the first system floppy. It also enters the name and address of NICK.V1 in the subdirectory NICK. (NICK is an entry in the subdirectory USERS which in turn is an entry in the root directory of the system floppy itself. All three directories and the virtual disk are on disk unit 1.)

The created virtual disk is of type DOS, with 13 sectors per track, so its size is automatically 455 sectors of 256 bytes, the size of a real mini-disk in DOS 3.2 or 3.2.1.

Anyone who enters the correct pathname and the password GOODSTUFF will be allowed to read, write or erase the virtual disk NICK.V1. Private access rights were not specified, so the defaults for a virtual disk (RWE) are in effect.

Users who do not know the private password are not allowed any access to NICK.V1.

2. CREATE /LANGUAGES/FRENCH/LESSON.3,T=P,SIZE=60

This example creates a virtual disk of type Pascal whose size is sixty 512-byte blocks. The virtual disk is called LESSON.3 and is listed in a subdirectory called FRENCH which in turn is listed in the root directory of a system floppy or hard disk called LANGUAGES.

Everyone has full access rights to this virtual disk without giving any password: they can read it and write on it, and if they have create, read, and delete rights for the subdirectory FRENCH, they can delete it.

3. CREATE BRIDGE,T=B,SIZE=10.S,D9,RW,SHR

This example creates a virtual disk called BRIDGE which is of type binary with a size of ten 256-byte sectors. Space is allocated on the physical unit specified in the current default directory, and the name and address of BRIDGE is entered in the current default directory. All users have full access rights. The disk is then mounted on virtual drive 9 for read/write access and shared usage.

4. CREATE STUDENTS,T=Y,PRVPW=SOCRATES

This example creates the directory STUDENTS on the physical unit specified in the current default directory and enters the name and address of STUDENTS in the current default directory. Only users giving the password SOCRATES are allowed full access rights; others are allowed to LIST the directory STUDENTS and read it when it is part of a pathname, but cannot, for example, make any new entries in the directory.

5. CREATE /COURSES/CHINESE/BEGINNING/LESSON1,T=P,PRVPW=MING,
GRPPW=PEKING,PUBACC=,GRPACC=R

This example is for a new Chinese course to be stored on a system 8-inch floppy called courses which has just been formatted by the Nestar formatting utility program (see System Manager's Manual). The root directory exists, but there are no other files on the floppy. In this example, the directories CHINESE and BEGINNING are implicitly created and the virtual disk LESSON1 is explicitly created. There is no public access to these three files. Users giving the private password MING are given the default private access rights: for the two directories they can create, delete, read, write and erase; and for the virtual disk they can read, write and erase. Users giving the group password, PEKING, can only read the three files.

DELETE

DELETE pathname

-deletes a file from system

Parameters and defaults:

pathname: For summary, including password conventions, see PATHNAME, filed alphabetically with commands in this chapter. For discussions, see Section 3.2

Error Messages

29,CURRENTLY MOUNTED OR DEFAULT DIRECTORY

Access Rights Required

E (ERASE) for virtual disk or directory being deleted
D (DELETE) for directory in which it is listed
R (READ) for all directories in pathname

Discussion

The DELETE command:

erases the specified virtual disk or directory and
removes its name and address from the parent directory

The DELETE will not be executed if: the file is currently mounted by anyone on the system, or the user does not have the required access rights, or it is a directory which is not empty, or it is the current default directory of any user.

Examples

DELETE ENGLISH.1A:OLDCHAP

This example erases (removes completely) the virtual disk ENGLISH.1A and removes (deletes) its name and address from the current default directory.

HELP [<subject>]

-displays list of commands or syntax of specified command or format of specified parameter

Parameters and Defaults

[<subject>]

null: lists commands

command word: gives syntax of specified command

parameter: gives format of specified parameter

CMD: gives syntax of special Pascal drive commands (SET DRIVE n[,n...] REAL|VIRTUAL [SLOTn][STATION \$nn], SET NFS SLOTn[STATION \$nn], SHOW NFS, SHOW DRIVES[NOCHECK]. This is available only when entering commands through CMD program.

Error Message

19,HELP OPERAND NOT FOUND

Discussion

The user is allowed to enter either ? or HELP when he needs additional information at a user-station. Options are (1) HELP, (2) HELP <command word>, (3) HELP <parameter>, (4) HELP CMD.

Example

If the user enters:

HELP

the following message is displayed:

ENTER 'HELP [<SUBJECT>]' WHERE SUBJECT IS: BLOAD, BSAVE, BRUN, CREATE, DELETE, LIST, LOCK, UNLOCK, MOUNT, UNMOUNT, PROTECT, SHOW, PATHNAME, PASSWORD, SET, NUMBER, SIZE, ACCESS, TYPE, USAGE, DRIVE, PROTECTION, RENAME, OR TIMESTAMP.

LIST

LIST [pathname] [,VERBOSE] [,NESTED]

-lists entries in directory

Parameters and Defaults

pathname: For summary, see PATHNAME, filed alphabetically with commands in this chapter. For discussion, see Section 3.2

[,VERBOSE]

if not present, just the names and file types are listed

if present, the names, types and sizes are listed

[,NESTED]

if not present, only entries in the directory addressed by the pathname or current default directory are listed

if present, the system lists

the addressed (or default) directory

all subdirectories whose names appear in the addressed directory

all subdirectories whose names appear in the listed subdirectories, etc. until all directories listed contain only names of virtual or system disks

(the files for each new subdirectory are indented several spaces)

Error Messages

109,ROOT DIR NOT SPECIFIED NO DEFAULT
102,FILE NOT FOUND

Access Rights Required

R (READ) access for each file in pathname

Discussion

This command can be used to list the files in a directory or a family of directories. It is possible to list either the names and types (no VERBOSE), or the names, types and sizes (VERBOSE).

If no pathname is given (e.g. LIST), then the current default directory is listed. If the NESTED or VERBOSE parameters are given when there is no pathname, they must be separated from the word LIST by both a space and comma. For example:

```
LIST ,NESTED
LIST ,VERBOSE
LIST ,NESTED,VERBOSE
LIST
```

Examples

1. LIST /MAIN

The system will display:

```
/MAIN,T=Y
  BOOT,T=B
  SYSTEM,T=Y
  DOS,T=Y
  PASCAL,T=Y
```

The T=Y indicates the files which are directories

2. LIST /MAIN,VERBOSE

The system will display:

```
/MAIN,T=Y,SIZE=3S
  BOOT,T=B,SIZE=11S
  SYSTEM,T=Y,SIZE=3S
  DOS,T=Y,SIZE=3S
  PASCAL,T=Y,SIZE=3S
```

The size is given in 256-byte sectors.

LIST

3. LIST /MAIN,NESTED

```
/MAIN,T=Y
  BOOT,T=V
  SYSTEM,T=Y
  ERROR,T=S
  HELP,T=Y
  HELP,T=S
  SHOW,T=S
  LIST,T=S
  BRUN,T=S
  LOCK,T=S
  MOUNT,T=S
  BLOAD,T=S
  BSAVE,T=S
  CREATE,T=S
  DELETE,T=S
  UNLOCK,T=S
  UNMOUNT,T=S
  PROTECT,T=S
  SET,T=S
  PATHNAME,T=S
  PASSWORD,T=S
  NUMBER,T=S
  SIZE,T=S
  PROTECTION,T=S
  ACCESS,T=S
  TYPE,T=S
  USAGE,T=S
  DRIVE,T=S
DOS,T=Y
  3.2.1,T=D
PASCAL,T=Y
  BOOT,T=P
```

T=S indicates the file contains file system information.

4. LIST /MAIN/USERS,VERBOSE,NESTED

This is left as an exercise for the reader.

LOCK lockname [,usage]

-locks "lockname" for exclusive (EXC) or shared (SHR) use

Parameters and defaults:

lockname:

any arbitrary name (except ALL)
(1-15 characters; no commas, slashes,
colons, control characters, carriage returns
or unprintable characters; imbedded blanks ok

Acceptable characters: 0-9, A-Z, !, ", #, \$,
%, &, ', (,), *, <, >, ?, [,], {, }, ^, ~,
@, _, |

usage:

EXC (exclusive)
SHR (shared)
Default: EXC

Error and Information Messages

40, ILLEGAL LOCK NAME
41, IN USE
42, IN EXC USE

Access Rights Required

None

Discussion

The LOCK command allows users at different terminals to cooperate in using system resources such as individual files or input/output devices. LOCKNAME is any arbitrary name agreed upon by more than one user. For example, if the lock is used to coordinate write access to a given DOS file, a program, before writing on that file, would issue a LOCK command. Then, if the return code is 0 indicating no other user-station is currently holding the specified lock, the program would proceed to update the file. If, however, the return code is 41 or 42 indicating the lock is currently held at another user-station, the program would go into a wait loop or perform some other task before reissuing the LOCK command.

NOTE: It is not recommended that programs enter a tight wait loop which includes a LOCK command. Instead, they should delay several seconds and then try again. This avoids tying up the

LOCK

network with frequent requests.

When using locks to protect a DOS or Pascal file, the user should consider the effect of buffering. When a disk write is issued, in some cases the information is transferred to the buffer but not to the disk. One way to make sure it all goes to the disk is to close the file. Buffering can also lead to problems if more than one lock is used for one file, say LOCKA for the first half and LOCKB for the second half; information at the boundary may not be handled correctly.

Therefore, the safest procedure to follow is:

1. Use only one lock for each file.
2. Close the file before issuing the UNLOCK.

Users who wish to use more than one lock per file or avoid closing the file after each write should have a detailed knowledge of exact buffering algorithms used by the current Apple DOS and Apple Pascal systems.

See Section 4.2 for a full discussion.

MOUNT pathname, drive [,usage]

-opens a file and establishes correspondence between it
and logical drive number

Parameters and defaults:

pathname: For summary (including password conventions), see PATHNAME,
filed alphabetically with commands in this chapter. For
discussion, see Section 3.2.

drive:

Dd, where d=number
(1-255 for DOS; 3-12 for Pascal)

usage:

RO (read only)
RW (read/write--must have write access)
Default: RO

EXC (exclusive)
SHR (shared)
UPD (RW usage for only one user at a time; other
users may mount the volume RO SHR)
Default: SHR, if RO
UPD, if RW
RO,SHR if no usage parameter given

Directories (T=Y) and system files (T=S)
may only be mounted for read only (RO) access.

Error Messages

30,DRIVE REQUIRED
31,IN USE
32,IN EXC USE
34,RW NOT ALLOWED ON DIRECTORIES
111,NO ACCESS FOR WRITE

Access Rights Required

R (read) for virtual disk and all
directories in pathname

W (write) for virtual disk if
being mounted RW

Discussion

The MOUNT command is analogous to inserting a real disk in a

MOUNT

floppy disk unit attached to a stand-alone Apple. A MOUNT command sets software pointers so that when a command is given to read or write on a given disk drive, it will address a virtual disk on the file server station.

MOUNT may be reissued on a currently mounted drive, either to change which file is mounted or to change usage modes, without intervening UNMOUNT.

Mounting a binary-type virtual disk is not recommended. Instead a BLOAD or BRUN should be used.

Examples

```
MOUNT /MAIN/USERS/CAROL/LETTERS,D9,EXC,RW
```

This command mounts a virtual disk called LETTERS, which is listed in a subdirectory CAROL, which in turn is listed in a subdirectory USERS, which in turn is listed in the root directory of a system floppy or hard disk called MAIN. The virtual disk will be mounted on drive 9; no other user can mount LETTERS until an UNMOUNT command is issued; and Carol can both read and write the files stored on LETTERS. If someone else has already mounted LETTERS, this MOUNT will not be executed; instead an error message

31,IN USE

will be returned.

This example does not specify any passwords. Therefore, it will be executed only if:

there is public read access to all directories and public RW access to the virtual disk LETTERS.

or

the default group or private password gives read access to all directories in the pathname and read and write access to the virtual disk LETTERS.

OFF

-initializes user-station: unmounts all MOUNTED files, unlocks all locks, resets any default directory and private or group passwords. BRUNS BOOT program.

Discussion

In order to increase over-all system efficiency, the OFF command should be given whenever a user is through using his Apple. This frees the memory and other resources assigned to the user-station.

Giving the file server command is equivalent to turning power off and on at the Apple or typing H at the Pascal command level in an Apple with an autostart ROM.

Note that when the power is turned off at a user station, all NFS files that are mounted remain mounted. Locks remain held and default directory, if any, remains set. If files are mounted EXC RW, for example, they will not be available to other users. It is the OFF command (which is automatically issued by the Network BOOT program) that unmounts all volumes. Thus, in order to free memory space and leave volumes available to other stations, it is necessary to use the OFF command in DOS or the H command in Pascal, or to turn the machine off and on again to unmount volumes.

PATHNAME

pathname: See Section 3.2 for a full discussion.

pathname: /unitname/filepath
or filepath

80 characters max.

An initial slash indicates first field is a unit name; no initial slash indicates current default directory is to be prefixed to pathname.

unitname: 1, 2, 5, or 6 [:password]
or name [:password]

filepath: name[:password] [/filepath]

name: name of disk unit, directory, or virtual disk.
(1-15 characters; may not include commas, slashes, colons, control characters, carriage returns or unprintable characters; imbedded blanks ok.)

Upper and lowercase can be used for any characters within a Network command and are equivalent.

For certain commands a null pathname is allowed:

SET DIR	sets current default directory to null
LIST	lists entries in current default directory
PROTECT	changes protection parameters for current default directory
SHOW PROTECTION	displays access rights for current default directory

pathname is used in BLOAD, BRUN, BSAVE, CREATE, DELETE, LIST, MOUNT, PROTECT, RENAME, SET DIR, SHOW MOUNTS, SHOW PROTECTION.

PROTECT pathname [protection]

-modifies passwords or access rights for a
virtual disk or directory

Parameters and defaults:

pathname: For summary, see PATHNAME, filed alphabetically
with commands in this chapter. For discussion,
see Section 3.2

[,protection] See Chapter 4 for a detailed discussion.

protection: protect item [,protection]

protect item: ,GRPPW=password
,PRVPW=password
,GRPACC=access rights (Default PUBACC)
,PRVACC=access rights (Default RWECD)
,PUBACC=access rights (Default R)

access rights: set of letters in any order from RWECD
(read, write, erase, create, and delete)

Error Messages

11,ILLEGAL PROTECTION PARAMETER
109,ROOT DIR NOT SPECIFIED; NO DEFAULT

Access Rights Required

Private password must be given.
No specific access rights required.

Discussion

The PROTECT command is used to change the password and access rights associated with a given directory or virtual disk. Anyone giving the private password is allowed to use the PROTECT command, regardless of the previously assigned access rights. If there is no private password, then anyone can issue the PROTECT command.

Normally access rights and passwords are set when a directory or virtual disk is created. The PROTECT command allows the user to change them without destroying and recreating the whole file.

A null pathname is allowed. It refers to the current default directory. If the null pathname is used, the comma must still be present, for example:

PROTECT

```
PROTECT ,PRVPW=APRICOTS,PRVACC=RE
```

Note that the PROTECT command does not change the access rights granted to users who have already mounted the virtual disk or directory, or who have a default directory set. The new access rights take effect only for subsequent MOUNT or SET DIR commands.

See Chapter 4 for a full discussion of protection.

Examples

```
PROTECT NANCY,PRVPW=VOLCANO,PRVACC=R,  
PUBACC=,GRPACC=
```

In this example anyone entering the password VOLCANO is allowed to read the virtual disk NANCY, and no one else has any access rights whatsoever. There are no public or group access rights because the protect items PUBACC= and GRPACC= specify no rights. No one can erase or write on this virtual disk until another PROTECT command is issued to assign such rights.

PROTECTION

[,protection] See Chapter 4 for a detailed discussion.

protection: protect item [,protection]

protect item: ,GRPPW=password
,PRVPW=password
,GRPACC=access rights (Default PUBACC)
,PRVACC=access rights (Default RWECD)
,PUBACC=access rights (Default R)

access rights: set of letters in any order from RWECD
(read, write, erase, create, and delete)

used in CREATE, PROTECT

RENAME

RENAME oldpathname, newname

-renames specified Network file

Parameters and defaults:

oldpathname: current pathname. For summary, see PATHNAME, filed alphabetically with commands in this chapter. For discussion, see Section 3.2

newname: last name in pathname is changed to newname

Error Messages

10,PARAMETER TOO LONG

103,BAD DELIMITER IN PATHNAME

Access Rights Required

C, D in parent directory; R for all directories in pathname

Discussion

Renames the specified Nestar file with 'newname'. The file to be changed is specified with a complete pathname (beginning with '/') or a partial pathname prefixed by the current default directory. Only the last name in the pathname is changed; do not supply a full pathname for 'newname'. E.g., to change /main/users/abc to /main/users/xyz enter

```
RENAME /MAIN/USERS/ABC,XYZ
```

A file that is currently mounted or set as a default directory may not be renamed. The name of a directory may be changed by supplying the pathname which identifies it. For example, to change /MAIN/USERS/... to /MAIN/PEOPLE/..., enter

```
RENAME /MAIN/USERS,PEOPLE
```

The name of a disk volume can be changed only with the offline file server utility FILDEBUG.

SET DIR pathname

-sets current default directory

Parameters and defaults:

pathname: For summary, see PATHNAME, filed alphabetically with commands in this chapter. For discussion, see Section 3.2

Error Messages

57,FILE IS NOT A DIRECTORY

Access Rights Required

Read access for all directories in pathname

Discussion

The SET DIR command acts differently depending on whether the specified pathname begins with a slash or not. If the pathname begins with a slash, the default directory is set equal to the specified pathname. But if the pathname doesn't begin with a slash, it is appended to the existing default directory. The last name of the specified pathname must be a directory.

In any subsequent commands, the current default directory is prefixed to pathnames which do not begin with a slash. Passwords may be required in a default directory, and the user is allowed to specify them in the SET DIR command or to use the default private and group passwords set previously by the PRVPW and GRPPW commands.

Access rights for the default directory are determined when the SET DIR is issued.

A SET DIR with no pathname clears the current default directory, i.e. sets it equal to null.

The SET DIR and SET password commands can be combined; for example:

```
SET DIR /MAIN,GRPPW=PINE,PRVPW=CEDAR
```

The delimiter between DIR and pathname can be either a blank or an equal sign.

The SHOW DIR command can be used to display the current default directory (excluding passwords).

SET DIR

Examples

1. SET DIR /1/ACCOUNTS/PAYROLL

This example sets the current default directory equal to the subdirectory PAYROLL which is listed in the subdirectory ACCOUNTS which is listed in the root directory of the 8-inch floppy disk mounted in the left drive of the system floppy disk. Now if the command is given to mount the virtual disk SALES

MOUNT SALES,D9,RO,SHR

it is equivalent to the command:

MOUNT /1/ACCOUNTS/PAYROLL/SALES,D9,RO,SHR

2. SET DIR SUNNYVALE

Because the pathname in this example does not begin with a slash it is appended to the existing default directory. If this command is issued after the previous example, the resulting default directory would be:

/1/ACCOUNTS/PAYROLL/SUNNYVALE

SET [GRPPW=password] [PRVPW=password]

-sets default group or private password

Parameters and Defaults

password: See Section 4.1.7 for further discussion.

a secret word used to limit access to files
(1-15 characters; may not include commas, slashes,
colons, carriage returns or unprintable characters;
imbedded blanks ok; first character can be any
acceptable character)

Error Messages

None specific to setting default passwords

Access Rights Required

None

Discussion

Having to enter a password for each file name takes time and makes it more likely that an unauthorized user will learn the password. Therefore, the Model A allows a user to enter two default passwords, one for private access and one for group access. Then if no password, or an incorrect password, is specified in a pathname, the system will see whether the private or group default password is correct, and, if so, will grant the corresponding access rights.

When a pathname is given, access will be granted under the following conditions:

The access desired is a public access right for the specified file; either no password or an extraneous password may be specified.

The access desired is a group access right for the specified file, and the correct group password is either specified in the pathname or stored as the default group password, or there is no group password.

SET password

The access desired is a private access right for the specified file, and the correct private password is either specified in the pathname or stored as the default private password, or there is no private password.

The SET password and SET DIR commands can be combined; for example:

```
SET DIR /MAIN,GRPPW=PINE,PRVPW=CEDAR
```

Examples

Suppose the teacher in charge of computer assisted mathematics decides he needs a password in order to write a particular disk. All courses have the same private password: OBVIOUS. So, at the beginning of his session, he enters:

```
SET DIR /MATH  
SET PRVPW=OBVIOUS
```

Then when he mounts the virtual disk for a given course, say ALGEBRA, instead of typing:

```
MOUNT /MATH:OBVIOUS/ALGEBRA:OBVIOUS,D9,RW,EXC
```

he can shorten it to:

```
MOUNT ALGEBRA,D9,RW,EXC
```

SHOW DATE

-displays the current date

Parameters and Defaults

None

Error Messages

None

Access Rights Required

None

Examples

SHOW DATE

The system displays the current date, for example:

DATE IS 04/01/81

SHOW DIR

SHOW DIR

-displays name of current default directory

Parameters and Defaults

None

Error Messages

64,NO DEFAULT DIRECTORY SET

Access Rights Required

None

Discussion

This command displays the pathname of the current default directory. Passwords supplied with a SET DIR are not displayed with SHOW DIR, but they are in effect. SHOW DIR always displays the directory pathname using a unit name rather a unit number.

Examples

SHOW DIR

The system displays the current default directory, for example:

/MAIN/USERS/ROOSEVELT

SHOW LOCK lockname

-lists all stations using
specified lockname

Parameters and Defaults

lockname

any arbitrary name (except ALL) agreed upon by cooperating users. Has the same restrictions as name: 1-15 characters; may not include commas, slashes, control characters, colons, carriage returns or unprintable characters; imbedded blanks ok; first character can be any acceptable character

Error Messages

2, NAME PARAMETER REQUIRED

Access Rights Required

None

Discussion

The SHOW LOCK lockname command lists the numbers of all stations currently holding the lock called lockname. The station numbers are given in hexadecimal (00 to FF).

Examples

SHOW LOCK PETER

This command will display all the station numbers which currently hold the lock PETER. For example:

04 2F 3C

SHOW [ALL] LOCKS

SHOW [ALL] LOCKS

-displays all locks set at user's own station (or all locks currently held in the file server, with station numbers of the holders).

Parameters and Defaults

None

Error Messages

None specific to SHOW LOCKS

Access Rights Required

None

Discussion

The SHOW LOCKS command lists all locks currently held by a user's own station. SHOW ALL LOCKS lists all locks currently held by any station, together with the numbers of the stations holding the locks.

Examples

1. SHOW LOCKS

This command displays for each lock currently held by a user's own station:

the lockname

whether usage is shared (SHR) or exclusive (EXC)

the number of user stations currently holding the lock

For example, if the following is displayed:

LOCKA	SHR	2
JILL	EXC	1
DAVID	SHR	1
PETER	SHR	3

it means that one other station currently holds LOCKA for shared usage; only the user's own station holds lock JILL for exclusive usage and lock DAVID for shared usage, and two other stations hold lock PETER for shared usage.

2. SHOW ALL LOCKS

This command displays for each lock currently held in the file server:

the lock name

whether usage is shared (SHR) or exclusive (EXC)

the number of user stations currently holding the lock

the station numbers of the holders

For example, if the following is displayed:

USER.JHM	SHR	1	28
PRINTER	SHR	2	1F
PRINTER	SHR	2	2C
USER.LJS	EXC	1	13

it means that station 28 currently holds lock USER.JHM for shared usage, while stations 2C and 1F hold lock PRINTER for shared usage, and station 13 holds lock USER.LJS for exclusive usage.

SHOW [ALL] MOUNTS [pathname]

SHOW [ALL] MOUNTS [pathname]

-gives information on virtual disks currently
mounted on the file server

Parameters and Defaults

None

Error Messages

131, UNIT NAME NOT FOUND

Access Rights Required

None

Discussion

(a) SHOW MOUNTS

The SHOW MOUNTS command displays for each virtual disk currently
mounted by the user's station:

its pathname

its drive number

whether it is read only (RO) or read/write (RW)

whether usage is shared (SHR) or exclusive (EXC), or
UPD

its type

its volume number

its size given in 256 byte sectors (S) for DOS and
512-byte blocks (B) for Pascal

SHOW MOUNTS always displays the directory pathname using a unit
name rather a unit number.

The NFS system uses some sectors for housekeeping information,
but these sectors are not counted when the SHOW MOUNT command
displays size. However, the LIST command displays a size which
may be one sector more than that shown by SHOW MOUNTS.

Examples

(a) SHOW MOUNTS

The system will display:

NAME	DRV	ACCESS	T	VOL	SIZE
/MAIN/PASCAL/LIB					
	5	RO SHR P	1	1200B	
/MAIN/USERS/EPS2					
	4	RW EXC P	1	280B	

Volume numbers are assigned by the user to each DOS virtual disk during the INIT command or with the VDOS CREATE utility; the default is 254. Pascal virtual disks are assigned number 1 by the NFS system.

(b) SHOW MOUNTS <pathname>

The SHOW MOUNTS <pathname> command displays the information specified by SHOW MOUNTS for the virtual disk with the given pathname if it is currently mounted by the user's station. Information is not given in nested form, so the pathname must be complete. That is, if the user has mounted /MAIN/USERS/JHM/1 and /MAIN/USERS/JHM/2, then the command SHOW MOUNTS /MAIN/USERS/JHM will return a null response; the user must ask explicitly for /MAIN/USERS/JHM/1.

(c) SHOW ALL MOUNTS

This form is reserved for use by the System Manager at the file server console.

(d) SHOW ALL MOUNTS <pathname>

This command returns information for the virtual volume specified by <pathname>, for every drive/station in the NFS network at which it is currently mounted. In addition, station numbers of those stations with the volume currently mounted and total number of stations using the volume are given (STN, USE):

SHOW [ALL] MOUNTS [pathname]

SHOW ALL MOUNTS /MAIN/PASCAL/LIB

NAME	DRV	ACCESS	T	VOL	SIZE	STN	USE
/MAIN/PASCAL/LIB							
	5	RO SHR P	1	1800B	20	3	
/MAIN/PASCAL/LIB							
	5	RO SHR P	1	1800B	07	3	
/MAIN/PASCAL/LIB							
	5	RO SHR P	1	1800B	1E	3	

SHOW PROTECTION [pathname]

-displays group, private and public access
rights for file addressed by pathname

Parameters and Defaults:

pathname: For summary, see PATHNAME, filed alphabetically
with commands in this chapter. For discussion,
see Section 3.2

Error Messages

64,NO DEFAULT DIRECTORY SET
110,NO ACCESS FOR READ

Access Rights Required

R (READ) access for every file in pathname

Discussion

This command can be abbreviated to

SHOW PROTECT [pathname]

It displays public, group and private access rights for the
specified file (or default directory if no pathname is
specified). Passwords are never displayed, but the existence of
passwords is indicated by "PW".

Example

SHOW PROTECT /MAIN/USERS/NSS/2

When this virtual disk was created, no usage rights were
specified, so that the system defaults are in effect. However, a
private password was set, so that access to the volume is
read-only unless the password is given.

PUBACC=R
GRPACC=R
PRVACC=RWE, PW

SHOW STATION

SHOW STATION

-displays the number of the user station

Parameters and Defaults

None

Error Messages

None

Access Rights Required

None

Examples

SHOW STATION

The system displays the hexadecimal user station number, for example:

STATION IS \$07

SHOW TYPE drive

-displays type of virtual disk mounted
on drive d

Parameters and Defaults

drive

Dd, where d=number
(1-255 for DOS; 3-12 for Pascal)

Error Messages

59,TYPE=SYSTEM
61,DRIVE REQUIRED
62,NOT CURRENTLY MOUNTED
65,TYPE=PASCAL
66,TYPE=DOS
67,TYPE=BINARY
68,TYPE=DIRECTORY
69,UNKNOWN TYPE

Access Rights Required

None

Discussion

This command is used to determine the type of a given file. Its main use is by BOOT programs which must decide whether to call Pascal or DOS when a given volume is booted.

Examples

SHOW TYPE D4

The system responds:

65,TYPE=PASCAL

SHOW TYPES

SHOW TYPES

-displays types of virtual disks mounted on drives 1-16

Parameters and Defaults:

None

Error Messages

None

Access Rights Required

None

Discussion

Returns the types of virtual disks (Y, P, D, B) currently mounted on drives 1-16. The information appears in a string, as shown in the example below. A dash, "-", indicates that nothing is mounted on the drive.

Pascal allows virtual mounts on drives 1-12; DOS allows virtual mounts on drives 1-255.

Example

SHOW TYPES

The system responds:

71,---PP---P-PP----

showing that the station has virtual Pascal disks mounted on drives 4, 5, 9, 11, and 12.

SHOW VOLS

-displays the currently active physical disk drive names and numbers.

Parameters and Defaults

None

Error Messages

None

Access Rights Required

None

Examples

SHOW VOLS

The system displays the current disk drives, for example:

DISK UNIT 1 IS /SPARE
DISK UNIT 5 IS /MAIN
0,OK

TIMESTAMP

TIMESTAMP

-returns encoded form of the current date and time

Parameters and Defaults

None

Error Messages

None

Access Right Required

None

Discussion

Returns encoded form of the current date and time. Format of the returned message is 70,ymmddhhmmssw where yy is current year (e.g., 81) mm is current month, dd is current date, and hhmmss are the current time. 'w' is the day of the week (Sunday is 1). If the clock/calendar card option is not installed, hhmmssw will all be zero, and yymmdd will be the date set by the System Manager.

UNLOCK lockname or UNLOCK ALL

-releases locked usage on a lockname
or on all locknames currently held

Parameters and Defaults

lockname

any arbitrary name agreed upon (except ALL)
(1-15 characters; no commas, slashes, colons,
carriage returns or unprintable characters;
imbedded blanks ok, first character any
acceptable character)

Error Messages

45,NOT HELD

Access Rights Required

None

Discussion

The UNLOCK command frees a lock on an agreed upon lockname, so that users at other terminals know it is all right now to access a file, external device, or other resource.

Examples

1. UNLOCK AMY

This command unlocks the lock called Amy.

2. UNLOCK ALL

This command unlocks all locks currently held by a user's own station.

UNMOUNT

UNMOUNT drive or UNMOUNT ALL

-cancels previous mount(s)

Parameters and Defaults

drive

Dd, where d=number (either \$hex or decimal)
(1-255 for DOS; 4,5,9-12 for Pascal)

Error Messages

35,DRIVE OR "ALL" REQUIRED
36,NOT CURRENTLY MOUNTED

Access Rights Required

None

Discussion

It is not necessary to do an UNMOUNT before mounting the same or a different virtual disk on a given drive. UNMOUNT is useful primarily to protect a virtual disk mounted with read/write access from accidental writes after it is no longer needed.

Warning: if a Pascal user issues a command to

UNMOUNT ALL

the Pascal system on that Apple will crash since there is no longer any booted volume on line. It will be necessary to turn power off and on to recover. Therefore, UNMOUNT ALL should never be given in Pascal, unless immediately followed by appropriate reMOUNTs.

Examples

UNMOUNT D9

This command resets the pointers which caused disk commands to read and write to a specified virtual disk whenever drive 9 was addressed.

Chapter 6

Using DOS with the Network File Server

DOS 3.2, 3.2.1, and 3.3 can run on user station Apples supported by the Network File Server. If there is no Language Card in a given Apple, either Integer BASIC or Applesoft BASIC can run, whichever is in the Apple ROM. If there is a Language Card, the Nestar BOOT program loads the type of BASIC not in ROM into the Language Card, so that both are available at all times. (Enter FP for floating point Applesoft or INT for Integer BASIC, using any DOS.)

6.1 Size and Number of Sectors

Unless the program VDOS CREATE is employed (Section 6.6), DOS will INIT virtual diskettes as if they were the same size as real 5 1/4 floppies. When creating DOS-type virtual diskettes the user specifies 13 or 16 sectors per track; for VDOS CREATE, 4 to 32 sectors/track and 18 to 50 tracks per volume.

<u>Operating System</u>	<u>Utility</u>	<u>Sectors/Track</u>	<u>Tracks/Vol</u>	<u>Total Sectors*</u>
DOS 3.2.1, 3.3	CREATE	13	35	455
DOS 3.3	CREATE	16	35	560
DOS 3.3	VDOS CREATE	4 to 32	18 to 50	72 to 1600

* plus one sector of file descriptor information

6.2 Drives

On a stand-alone Apple, DOS allows only drives 1 and 2 to be specified, but the Cluster/One system allows any drive number from 1 to 255. To enable this capability, the Nestar BOOT program POKES the number \$FF (255) into the DOS byte containing the maximum drive number.

If an Apple user-station has no standard Apple disk control card, using drive numbers greater than 2 will cause no problem; but if there is a standard Apple disk control card and a user erroneously gives a drive number greater than 2 in a command to a real mini-disk, DOS will not catch the error and results are unpredictable.

6.3 Using Immediate Execution Mode in DOS

See Section 2.5 for a hands-on tutorial. The following is a

brief summary:

1. DOS presents the prompt, > for INT or] for FP.
2. Enter PR#n where n is the number of the slot containing the network interface card, typically 6:

PR#6

3. DOS again presents > or]
4. Enter an @ sign followed by a file server command:
e.g.

@SHOW DATE

5. A return code number and message is displayed on the screen; this may be either an error message or an information message, e.g.

0,OK
1,ILLEGAL COMMAND
66,TYPE=DOS

6. Steps 3, 4, and 5 may be repeated indefinitely.

6.4 Using File Server Commands within a BASIC Program

Different BASIC commands are used to communicate with the file server, depending on whether Integer or Applesoft BASIC is being used. The differences are due to the fact that CHR\$ is not available in Integer BASIC and the INPUT command behaves somewhat differently in the two languages.

6.4.1 Applesoft BASIC

The following Applesoft program segment sends a file server command to the disk-server and reads the code and message it sends back. It is assumed that DOS is present and operational.

```
10 CD$="any Network Command"
```

Sets the string variable CD\$ equal to a string consisting of a file server command

```
20 PRINT CHR$(4);"PR#6"
```

CHR\$(4) is a ctrl-D. It signifies that the next field is a DOS command. "PR#6" enables the network card to look at all subsequent print statements to see whether they contain a

Network command to ship to the file-server.

```
30 PRINT CHR$(14),CD$
```

CHR\$(14) is a ctrl-N. It signifies that the next field is a file server command.

```
40 PRINT CHR$(4); "IN#6"
```

CHR\$(4) again signifies a DOS command. "IN#6" tells the following INPUT command to take its input from slot 6 rather than from the keyboard.

```
50 INPUT NN,NN$
```

The disk-server sends its return code (NN) and message (NN\$) to the network card in slot 6. After reading the last character from slot 6, the INPUT command resets the input slot so the next input will come from the keyboard.

Note: The "PR#6" and "IN#6" assume the network card is in slot 6. If the card were actually in slot 7, the commands would be "PR#7" and "IN#7". The exact slot must be specified.

See Section 2.7 for a greeting program which uses file server commands with Applesoft BASIC.

6.4.2 Integer BASIC

The same program segment written in Integer BASIC follows. This program, unlike the Applesoft program above, prints an error message if the file server command cannot be executed. It is assumed that DOS is present and operational.

```
10 DIM RET$(50), CD$(120)
```

```
20 N$="":REM THERE IS AN INVISIBLE CTRL-N BETWEEN THE QUOTES
```

```
30 D$="":REM THERE IS AN INVISIBLE CTRL-D BETWEEN THE QUOTES
```

```
40 CD$="Any file server command"
```

```
50 PRINT D$;"PR#6"
```

D\$ is a ctrl-D; it signifies that the next field is a DOS command. PR#6 enables the network card to look at all subsequent print statements to see whether they contain file server commands.

```
60 PRINT N$;CD$
```

N\$ is a ctrl-N; it tells the network card that the following field is a file server command to be sent to the disk server.

```
70 PRINT D$;"IN#6"
```

D\$ again signifies a DOS command. "IN#6" tells the following INPUT command to take its input from slot 6 rather than from the keyboard.

```
80 INPUT RET$
```

The file-server sends its return code number and message (separated by a comma) to the network card in slot 6. After reading the last character from slot 6, the INPUT command resets the input slot so next input will come from the keyboard.

```
90 IF RET$#"0,OK" THEN PRINT "ERROR ";RET$
```

This command prints any return code and message except "0,OK"

6.5 Initializing a Virtual Disk in DOS

See Sections 2.6 and 2.7. The examples given there are for Applesoft BASIC. If Integer BASIC is to be used, use the above example to modify the CHR\$ and INPUT commands and add a dimension statement for the strings.

Note: You must always INIT a 13-sector virtual diskette to be used under DOS 3.2.1. You may CREATE (NFS command) and then INIT (DOS command) a 16-sector virtual diskette under DOS 3.3, or run VDOS CREATE (a Nestar utility program run from DOS; Section 6.6), which will create and initialize the disk for you. However, it is possible to both read and write these two virtual diskette formats from either DOS using the Network file server. You need not, therefore, convert all DOS 3.2.1 virtual diskettes to DOS 3.3 in order to use DOS 3.3. This is not true for real Apple diskettes.

6.6 Creating and INITING a Variable-Sized Virtual Diskette Using VDOS CREATE

VDOS CREATE allows DOS 3.3 users to create virtual DOS volumes of variable size, from 72 sectors (18 Kbytes) to 1600 sectors (400 Kbytes) (K=1024 bytes). The user is given a menu of eight parameters that can be varied within programmed limits that will affect total size of volume,

volume number, tracks per volume, sectors per track, catalog entry capacity, and boot capability -- all of which, in turn, affect the usable number of sectors after INIT, the ninth displayed menu parameter. VDOS CREATE allows the user to leave the DOS code off the newly created volume if it is not needed, saving approximately 37 sectors for other use.

Operation:

- Boot into MAIN/DOS/3.3 on a 48K Apple][plus or Apple][with Applesoft ROM card or 16K RAM card (language card).
- RUN VDOS CREATE
- Use the spacebar key to select one of the top eight menu items. VOLUME: is the pathname for the volume to create; other prompts are self-explanatory.
- Change the selected parameter by typing in letters or numbers. VDOS CREATE will adjust when necessary other parameters in the menu and perform bounds checking.
- Press a ? key to display and enter parameters' default.
- Press ESC key when parameter specification is complete.

VDOS CREATE will then:

- attempt to CREATE, MOUNT and INIT the user-named volume using the parameter specifications.
- trap any NFS or DOS 3.3 errors that will prevent program termination.
- cause a user-specified, non-bootable volume to print, "NO DOS TO BOOT" when a boot is attempted.
- save a dummy HELLO program on the VDOS CREATE created volume.

Parameter Limits:

	Min	Max
Volume number	1	254
Total sectors on volume	72	1600
Sectors per track	4	32
Tracks per volume	18	50
Catalog entries	7	217

Note that once VDOS CREATE has been run, the DOS INIT command is disabled until the user reboots.

Use <escape><escape> to abort VDOS CREATE while in the menu.

VDOS CREATE is written in Applesoft and 6502 machine code. VDOS CREATE does not support real disks (5.25 inch floppy).

6.7 Using DOS From Real Diskettes

If a real diskette containing DOS is copied to a virtual volume, the DOS on that volume remains unchanged. However, if the virtual volume is used to boot the user station, the Network BOOT program will make a five-byte patch to DOS in RAM so that it is compatible with the file server network.

To copy entire volumes, real to virtual or virtual to real (using Nestar DISK COPY), or selected files from those volumes (using FIDMORE and FID), you must have booted from a virtual DOS volume of the same size (that is, 13 sector or 16 sector) as the volumes to be copied from and onto.

Chapter 7

Using Pascal with the Cluster/One System

The Cluster/One Model A supports user stations running with Apple Pascal Versions 1.0 and 1.1 (U.C.S.D. Version II.1). In order to use Pascal on a given user-station, the Apple must contain a Language Card in slot 0.

Essentially, the Pascal Operating System runs in the Cluster/One system just as it does with a stand-alone Apple. However, there are extensions, which will be covered in this chapter:

- virtual diskettes of variable size
- virtual and real disk drives
- issuing file server commands with the CMD program
- issuing file server commands within a user's program.

7.1 Virtual Diskettes of Variable Size

Size is an optional parameter when CREATEing a Pascal-type virtual diskette. If no size is specified, each Pascal-type virtual diskette is the size of a real mini-disk, i.e., 280 blocks (each 512 bytes long).

Size may be specified with the CREATE command using the parameter:

[,SIZE=b]

where b is the number of 512 byte blocks. The maximum size is limited by the size of the disk unit on which the virtual diskette will be stored and by limitations in the Pascal operating system. The number of blocks for a Pascal-type virtual diskette must be:

>0	(minimum)
<=1200	(approximate maximum that will fit on an 8-inch system diskette)
<=32767	(max for system hard disk)

NOTE: The system also allows size to be specified in units of:

C	(characters--1 byte each)
S	(sectors--256 bytes each)
B	(blocks--512 bytes each)
blank	(blocks--512 bytes each)
K	(kilos--1024 bytes each)

by appending the appropriate character at the end of the size specification (e.g., SIZE=280B). Numbers may be given in \$hex or decimal. For details, see the CREATE command writeup in Chapter 5.

7.2 Drives

Each user-station operating under Pascal has 10 mini-disk drives (volume numbers 3-12) that can be either "real" or "virtual". Real drives are local, private mini-disks attached directly to the user-station; virtual drives are volumes that reside on an 8-inch system floppy or system hard disk connected to a (shared) file server station.

The BOOT program, part of the Cluster/One System software (see Appendix B), sets the drive status of each drive when a Pascal virtual diskette is booted at an Apple user-station. Whether a drive is set by BOOT to be real or virtual depends on the presence or absence of network interface cards or standard Apple DOS 3.2 or Pascal Disk II controllers in certain slots. At power-on the drive-status for each drive is as follows:

<u>Drive</u>	<u>Drive-Status</u>
3	Real. May be set virtual by user using NESTAR.START program (Section 8.2.2) or CMD program (Section 7.3.4).
4 and 5	Always virtual to booted slot (typically 6 but can be 1-7).
6-8	Real. May be set virtual by user using NESTAR.START program (Section 8.2.2) or CMD program (Section 7.3.4).
9 and 10	Real if slot 4 contains a standard Apple disk controller Virtual to slot 4 if slot 4 contains a network interface card Virtual to booted slot if slot 4 is empty
11 and 12	Real if slot 5 contains standard Apple disk controller Virtual to slot 5 if slot 5 contains a network interface card Virtual to booted slot if slot 5 is empty

The convention for setting the initial drive-status (as described above) will usually provide acceptable defaults for a wide variety of user-station configurations. In the normal case, where the network card is in slot 6 and slots 4 and 5 are empty, the drives 4, 5, 9, 10, and 11 are all virtual to slot 6. The CMD program (described in the following section) can be used at any time to display and/or change the the current drive-status values.

BOOT boots from the highest numbered slot containing a network card (normally 6) and looks for second and third network cards in slots 4 and 5.

If multiple network interface cards are present, and the secondary (or tertiary) nets are not running, Pascal will try to read disk labels from these drives, and may hang waiting for a response. In this situation typing ctrl-shift-N will permit things to proceed. In fact, if only one network interface card is present, and that net is temporarily not running, entering ctrl-shift-N will permit local processing to continue at the user-station. However, if the user wishes to reboot, it is necessary to press RESET.

7.3 The CMD Program. Entering File Server Commands From The Keyboard.

The CMD program is an interactive Pascal program supplied by Nestar, which allows a user to enter file server commands at the keyboard. CMD is generally made available as a CODE file in the default library volume automounted on drive 5 by BOOT. If CMD is not found there, the System Manager will know where it is installed.

The CMD program sends each command, as entered, to the file-server (except for the local commands described below). The file-server executes the command. If there is an error, a numeric error code and an error message will be displayed. A complete list of error messages is found in Appendix A.

CMD then redisplay on the user station screen its prompt (:) for another file server command to be entered. The cycle repeats until "Q" <return> or <esc><return> is entered; then control returns to the Pascal command level.

The following print-out shows a sample run. User responses are shown in lower case to distinguish them from information displayed by Pascal and CMD. (Actually, commands can be entered in either upper or lower case, and output from

PASCAL

Pascal on an Apple with upper/lower case capability is usually upper and lower case.)

```
COMMAND: E(DIT,R(UN,F(ILE,C(OMP,L(INK,X(ECUTE,A(SSEM,D(EBUG,?  
x
```

```
EXECUTE WHAT FILE? #5:cmd
```

```
ENTER NETWORK COMMAND
```

```
: create lunch,t=p  
109,ROOT DIR NOT SPECIFIED; NO DEFAULT
```

```
: SET DIR /MAIN/PASCAL
```

[No message in response to a command indicates that the command was successfully executed.]

```
: SHOW DIR  
/MAIN/PASCAL
```

```
: create lunch,t=p
```

```
: Q (return)
```

```
COMMAND: E(DIT,R(UN,F(ILE,C(OMP,L(INK,X(ECUTE,A(SSEM,D(EBUG,?
```

In addition to the commands that it sends to the file server, CMD responds to a number of local commands that it executes itself. These commands are concerned with displaying and setting the drive status (real or virtual) and network interface card slot number associated with each Pascal drive.

7.3.1 SHOW DRIVES [NOCHECK]

This command displays the status of all the Pascal units. The information displayed consists of the following:

- the unit number.
- a "#" character if the unit is a disk. (same convention as Pascal filer's "V" command.)
- volume name.
- number of blocks on device (disks only).
- REAL or VIRTUAL indication. If VIRTUAL then the "virtual channel" (i.e., the slot of the network card and station number of the file server) is displayed.

The SHOW DRIVES display has the following format:

DRV	T	NAME	BLKS	STATUS
1		CONSOLE:		REAL
2		SYSTEM:		REAL
3	#	-----		VIRTUAL:
4	#	ABC1:	800	VIRTUAL: SLOT6,STN=\$FE
5	#	PASCAL:	1800	VIRTUAL: SLOT6,STN=\$FE
6	#	-----		VIRTUAL:
7	#	-----		VIRTUAL:
8	#	-----		VIRTUAL:
9	#	-----		VIRTUAL:
10	#	-----		VIRTUAL:
11	#	-----		VIRTUAL:
12	#	ABC2:	800	VIRTUAL: SLOT6,STN=\$FE

In this example, the user has virtual volume ABC1 mounted on drive 4 and virtual volume ABC2 mounted on drive 12. Drives 3 through 12 have been set to virtual. The file server station number is \$FE. The shared library virtual volume is mounted on drive 5.

Normally SHOW DRIVES will read in all the volume names to make sure the information is current. However, the NOCHECK option can be used to suppress this check.

7.3.2 SHOW NFS

This command displays the default "virtual channel" (consisting of a network card slot number and file server station number) to which file server commands are currently being sent.

7.3.3 SHOW SLOTS

This command displays the Pascal units connected to the network card(s) in each slot. In addition, the station number of the network card(s) is displayed. For example:

```
:show slots
SLOT TYPE      USAGE
  6 NET ($21) NFS,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
```

Here, the user station is the same as in the SHOW DRIVES example above. Drives 3-12 have been set to virtual through slot 6. The station number is \$21 and the station is attached to only one network.

7.3.4 SET DRIVE[S] n[,n,...] REAL|VIRTUAL, [SLOT n] [STN \$nn]

This command sets the drive number (or numbers) given to the status specified. If REAL is specified, then the real Apple

PASCAL

disks are used (if present). If VIRTUAL is specified then the drive is a virtual drive connected to a NFS. The SLOT and STATION number of the NFS can be specified, if the NFS is other than the default SET NFS.

Drive numbers in the range 3 to 12 are available. For example,

```
:set drives 3,6 real
:show drives
```

DRV	T	NAME	BLKS	STATUS
1		CONSOLE:		REAL
2		SYSTEM:		REAL
4	#	ABC1:	800	VIRTUAL: SLOT6,STN=\$FE
5	#	PASCAL:	1800	VIRTUAL: SLOT6,STN=\$FE
7	#	-----		VIRTUAL:
8	#	-----		VIRTUAL:
9	#	-----		VIRTUAL:
10	#	-----		VIRTUAL:
11	#	-----		VIRTUAL:
12	#	ABC2:	800	VIRTUAL: SLOT6,STN=\$FE

Here, a user station has used SET DRIVES to change drives 3 and 6 at his station to real. These drives are now available for use with real floppy disk drives. When set real, drives 3 and 6-8 are invisible to Pascal.

7.3.5 SET NFS [SLOT n] [STN \$nn]

Sets the default "virtual channel" for NFS commands issued by CMD or by subsequently run programs to the values specified. The initial default NFS channel is the slot booted from (usually slot 6) and station \$FE.

7.3.6 Other CMD Inputs

The CMD program also accepts the following inputs:

; <anystring> Ignored as a comment.

PRINT <anystring> Displays the text on the console.

command; command; ... Multiple CMD commands are allowed on the same input line if they are separated by a semi-colon.

%filename Causes commands to be read in from the Pascal text filename specified.

NOECHO command Commands from a file (initiated by %filename) are usually displayed when they are executed; NOECHO suppresses this.

7.4 Using File Server Commands Within a Pascal Program

NESTAR provides Pascal "units" which allow the Pascal user to write programs that issue commands to the file server. The unit library distributed is called

NETWORK.LIBRARY

and it contains the following units:

CMDUNIT - a Pascal regular unit that allows programs to issue file server or local commands (i.e., any command accepted by the CMD utility program). In fact, this is the unit that the CMD utility program uses. This unit requires the unit NETUNIT, also contained in NETWORK.LIBRARY.

NETUNIT - a Pascal regular unit that allows programs to issue commands to the file server (local commands are NOT accepted here) and optionally gets responses back for processing by the program. In addition, NETUNIT contains routines that allow direct station-to-station communication. These routines are not described here, but a description can be found in Nestar Technical Note #7, NETUNIT.

NETWORK.LIBRARY may be installed in SYSTEM.LIBRARY at your local installation. If not, the Pascal program must use the \$U option to specify the location of the unit library to the compiler. During linking, the same library name must also be given.

A program using CMDUNIT and/or NETUNIT must have the following "USES" in the source of the program:

```
USES (*$U network-library-name*) NETUNIT,CMDUNIT;
```

Note the order: NETUNIT must come before CMDUNIT for the declarations to compile without errors.

If the program uses only NETUNIT, then "USES" in the source of the program can be simplified to:

```
USES (*$U network-library-name*) NETUNIT;
```

PASCAL

7.4.1 CMDUNIT

If the user wants to execute file server or local commands (just as the CMD utility program does), then the Pascal procedure CMD or CMDLOOP in CMDUNIT should be used.

The CMD procedure has the form:

```
CMD (command, options)
```

where:

COMMAND is a string variable whose value is the file server command or local command to be executed.

OPTIONS is an integer variable which must be set to:
0 - don't echo command if error.
1 - do echo command if error.

Examples:

```
CMD('show mounts',0);  
CMD('show drives',0);  
CMD('mount /MAIN/A,D11; MOUNT /MAIN/B,d12',1);
```

For programs in which the user enters file server or local commands from the keyboard until the user types QUIT or <escape>, the CMDLOOP procedure is provided. It has the form:

```
CMDLOOP (PROMPT);
```

where:

PROMPT is the string variable whose value is the prompt to be issued to the user.

Examples:

```
CMDLOOP(': ');  
CMDLOOP('N(fs) command or QUIT: ');
```

The following program, COMMAND, demonstrates the use of CMDLOOP to execute file server commands. This is, in fact, the complete source listing of the CMD utility program:

```
PROGRAM COMMAND;  
  
USES (*$U MODAOBJ:NETWORK.LIBRARY *) NETUNIT,CMDUNIT;  
  
BEGIN
```

```

      CMDLOOP(' : ')
    END.

```

7.4.2 NETUNIT

To execute a file-server command and get the return code back for evaluation by the program, use the NFSCMD procedure. The procedure has the form:

```
RC := NFSCMD (NIL,COMMAND,RESPONSE);
```

where:

RC	is an integer representing the "return code" sent back by the file server (see Appendix A for a list of return codes and their meanings). A return code of 0 means no errors.
NIL	is a "virtual channel" pointer. Specify NIL for the default (and perhaps only) file server. See Nestar Technical Note 7, NETUNIT for a discussion of multiple file servers and virtual channels.
Command	is a string variable whose value is the file server command to be executed.
RESPONSE	is a string "VAR" variable whose value is set to the last response line sent back by the file server. If the command has been executed successfully, then the last response is usually "OK". Otherwise, the last response contains the error message text.

All intermediate response lines are written to the screen.

Note: The NFSCMD procedure may wait an arbitrarily long time for the network to become available so that the command can be sent to the file server. This wait may be inappropriate if the program needs to check for other events frequently (such as key presses). For this reason, there is also a NFSCMD0 procedure. The procedure has the form:

```
RC := NFSCMD0 (NIL,COMMAND,RESPONSE);
```

which is identical to NFSCMD, except that the procedure will not retry if the network is not immediately available. The calling program should retry (and check for other events). For example:

```
repeat
```

PASCAL

```
RC := NFSCMD0 (NIL,COMMAND,RESPONSE);  
until RC <> 98; (*repeat until not network error  
(busy)*)
```

To execute a file server command and get all the responses back for evaluation, (including those usually displayed on the screen without program intervention), use the NFSCMD1 procedure. The procedure has the form:

```
RC := NFSCMD1 (NIL,COMMAND,RESPONSE,NEWCMDFLAG,DONEFLAG);
```

where:

RC	is an integer representing the "return code" sent back by the file server.
NIL	is a "virtual channel" pointer. Specify NIL for the default (and perhaps only) file server. See Nestar Technical Note 7, NETUNIT for a discussion of multiple file servers and virtual channels.
COMMAND	is a string variable whose value is the file server command to be executed when NEWCMDFLAG is set to TRUE. When NEWCMDFLAG is set to false it is ignored.
RESPONSE	is a string "VAR" variable whose value is set to the next (or first) response line sent back by the file server.
NEWCMDFLAG	is a boolean variable whose value is TRUE when a command is initially sent to the file server. It is FALSE on successive calls which get the next response lines.
DONEFLAG	is a boolean "VAR" variable. Its initial value is ignored and it is set to TRUE if this is the last response line and FALSE if there are more response lines.

For example, a program fragment might contain:

```
CMD := 'file server command';  
RC := NFSCMD1 (NIL,CMD,RSP,TRUE,DONE);  
WHILE NOT DONE DO BEGIN  
    ...process RSP (note: RC not valid until DONE = TRUE)  
    RC := NFSCMD1(NIL,CMD,RSP,FALSE,DONE);  
END;  
    ...process final RSP and RC...
```

7.4.3 A Sample Program

The following program, MINICMD, demonstrates the use of NFSCMD to send commands to the file server with a Pascal program. The MINICMD program will get an input line from the user and send it to the file server for execution.

MINICMD differs from CMD in the following ways:

1. Both the code number and reply returned by the file-server are displayed on the console.
2. The commands referring to the user's own drives (SHOW DRIVES, SHOW SLOT, SET Sn, and SET Dd, drive-status) are not available in MINICMD. They are executed by the CMD program, using CMDUNIT, not by the file-server.

```

PROGRAM MINICMD;

USES (*$U PASCAL:NETWORK.LIBRARY *) NETUNIT;

VAR
  RC: INTEGER;
  CMD, REPLY: STRING;

BEGIN
  WRITELN ('Mini Command Program');
  REPEAT
    WRITE ('NFS COMMAND? '); READLN (CMD);
    IF LENGTH (CMD) <> 0 THEN BEGIN
      RC := NFSCMD (NIL, CMD, REPLY); (* Send command to
                                      default NFS *)
      WRITELN (RC, ', ', REPLY)
    END;
  UNTIL LENGTH (CMD) = 0; (* All done if nothing typed *)
  WRITELN ('Ciao.')
END.

```

A sample execution of the MINICMD program :

```

Mini Command Program
NFS COMMAND? show date
DATE IS 03/10/81
NFS COMMAND? foo
1, ILLEGAL COMMAND
NFS COMMAND?
Ciao.

```

7.5 Multiple File Servers

Cluster/One systems may have more than one file server on the same network. Multiple file servers are only accessible to user stations operating in the Pascal environment. In addition, user stations may be attached to more than one network.

A user station identifies file servers by the number of the slot containing the network interface card for the network that the file server is attached to, and by the station number for the file server on that network.

User stations have two kinds of communications with file servers: users may issue commands to the NFS, and may issue I/O requests for virtual drives. A user station may be configured so that it has virtual disks mounted on a number of different file servers at the same time. The Pascal system can then be used to transparently access any combination of these virtual disks, just as if they were real disks, regardless of what actual file server station they are stored on.

The commands SET NFS and SET DRIVES (Sections 7.3.4 and 7.3.5) allow the user to configure his Pascal 'drives' to correspond to virtual disks on various file servers. These commands can set drives 3-12 to be real or virtual, to define which file server the drives are connected to, and to set a default "virtual channel" that will be used when NFS commands are issued.

There are many interesting and complex system configurations that can be constructed using multiple file servers, but they are all built upon the same basic primitive operations. As an example, suppose that it is necessary to manually transfer a file within a Pascal volume from one file server to another on the same network.

Assume that, in addition to the primary file server on station \$FE, the system manager has installed another file server on station \$FD. The following procedure will transfer a file called "XREF.TEXT" from Pascal volume "/MAIN/USERS/LJS/1" on the primary file server (\$FE) to the Pascal volume "/MAIN/ARCHIVE/PVOLS" on the second file server (\$FD):

- (1) Boot a Pascal volume from the primary (\$FE) file server.
- (2) From the command level of the Pascal system, X(ecute the CMD program. Issue the following commands:


```

SET DRIVE 11 VIRTUAL          (Set D11 virtual to $FE)
MOUNT /MAIN/USERS/LJS/1,D11   (Mount the source volume)
SET NFS STATION $FD           (change default NFS)
MOUNT /MAIN/ARCHIVE/PVOLS,D12,UPD (Mount the destination
                                volume on NFS $FD)
SET DRIVE 12 VIRTUAL          (Set D12 virtual to the default
                                NFS, which is now station $FD)
SET NFS STATION $FE           (Restore the default NFS for
                                future commands)
QUIT                           (Exit from CMD)

```

You can use the "SHOW DRIVES" and "SHOW NFS" commands to see the status of drives and the NFS command channel.

- (3) Use "F" to go to the Pascal filer, and T(ransfer from "#11:XREF.TEXT" to "#12:\$". The file will be copied from the \$FE file server to the \$FD file server.

This example is "manual" in the sense that the commands are typed interactively. The same operations can be performed from within a program by using the CMDUNIT unit provided with the system to issue the commands, and then using Pascal I/O of any kind to transfer the data. Other interesting additions, such as using symbolic or indirect names instead of the file server station numbers, can be added at the same time.

7.6 Initializing a Virtual Diskette in Pascal

Suppose that a brand new user wants to write new Pascal programs or run existing ones on the Cluster/One Model A system. What must he do to get started?

In general, he (or the system manager) must:

Boot an existing Pascal virtual disk.

CREATE a Pascal-type virtual disk using CMD.

MOUNT it on drive 3, or 6-12 for RW access. (The drive must be virtual, not real. Remember that the default for drives 3, 6, 7, and 8 is real.)

PASCAL

Use the Filer to zero its directory (See Apple Pascal Operating System Reference Manual, p.50).

Boot the new virtual disk.

Use the Editor to create a STARTCMD.TEXT file of startup commands on the virtual disk. The commands will be executed automatically whenever the disk is booted.

Use the Filer to transfer files from a real disk inserted in a mini-disk attached to the user station, if desired.

Use the Editor, Compiler, and Linker to create Pascal programs, just as on a stand-alone Apple.

For example, say that an educator has a set of arithmetic courses which he previously ran on a stand-alone Apple. Now he wants to present these same courses on the Cluster/One Model A.

Let's assume there is a system 8-inch floppy called COURSES and that he wants to CREATE a virtual disk called ARITHMETIC on it. He would give the CREATE command:

```
CREATE /COURSES/ARITHMETIC,T=P,D9,RW
```

(Assume that no protection is desired and that the size of a real diskette is satisfactory. CREATE is used here to mount as well as create the virtual disk.)

Then using the Pascal FILER, he would zero the directory on the virtual disk, and transfer files from the real disk to his virtual one.

Chapter 8

User Station Utilities

8.1 Introduction

In addition to VDOS CREATE (Chapter 6), CMD (Chapter 7), and the network BOOT program (Appendix B), Cluster/One Model A provides a number of utility programs for use at the individual network stations. These programs are distributed as part of the standard File Server system. Before attempting to execute them, ask your System Manager for the names of the volumes on which they reside at your installation. A form has been provided at the back of this manual for use in recording these names and other volume/file information.

Pascal User Station Programs

SETPROFILE is used by the System Manager to create a file of autoboot defaults for an individual station. Once this file has been created, the defaults will be used by the file server BOOT program every time the station is powered on. The user can specify to the System Manager which volume is to be prompted for or mounted automatically at power on, and the access rights to be assigned to that volume. Any startup program on the volume will then be run.

NESTAR.START executes a file of commands on the user's boot volume at Pascal initialization time.

FILEINFO allows the user to display on his console internal information about files on an NFS disk.

TDELETE allows the Pascal user to delete entire subtrees of the file system.

NETLINK allows the user to access the Cluster/One network when Pascal was booted from standard Apple floppy disks and not over the network.

DIRLIST writes a Network subdirectory to a Pascal file or to the console.

DOS User Station Utilities

- SETPROFILE is used by the System Manager to create a file of autoboot defaults for an individual station. Once this file has been created, the defaults will be used by the file server BOOT program every time the station is powered on. The user can specify to the System Manager which volume is to be mounted automatically and booted from at power on, and the access rights to be assigned to that volume. Any "hello" program on the volume will then run.
- NESTAR DISK COPY allows the user to copy the contents of an entire real or virtual disk (13 or 16 sectors), to another real or virtual disk, using any combination of Apple drives and network virtual drives.
- FILEINFO allows the user to display on his console internal information about files on an NFS disk.
- FIDMORE allows a user to load and modify a copy of Apple DOS 3.3 FID to accept drive numbers from 1 to 9 instead of just 1 to 2.
- VDOS CREATE allows a user to CREATE and initialize a variable size DOS 3.3 volume. See 6.6 for operation.

8.2 Pascal User Utilities

8.2.1 SETPROFILE Utility Program

SETPROFILE is a Nestar utility program run by the System Manager at a user station. The program creates a network volume that contains the name of the volume to be booted at that (or some other) station when the power is turned on.

The user can specify to the System Manager whether the volume should be mounted automatically or only prompted for, and what the access rights to the volume are to be. If the volume is mounted automatically, any startup program on it will then run.

SETPROFILE is explained in more detail in Section 5.8 of the System Manager's Manual.

START

8.2.2 NESTAR.START Command Program

NESTAR.START is the program that automatically runs in the user station when the boot volume is mounted. NESTAR.START first checks the booted disk for a file named SYSTEM.STARTUP. If present, it is chained to. Otherwise, NESTAR.START reads a file from the booted disk called STARTCMD.TEXT and interprets each line as a command.

Note that in the Cluster/One System, users do not need to have SYSTEM.PASCAL or SYSTEM.APPLE on their boot volumes to run in the Pascal environment. Instead, all stations can use the Nestar-provided equivalent code files on a shared library disk. See Appendix B (BOOT Program) and the ZAPPASCAL Utility Program in the System Manager's Manual (Section 5.7) for more details.

The NESTAR.START program reads a file from the booted disk called STARTCMD.TEXT and interprets each line as a command. The format of each command is:

```
<command>      ::=      <verb><one-or-more-blanks>
                        <optional-operand>
```

The legal <verb> <operand> sequences are as follows:

; <anystring> Ignored as a comment.

PRINT <anystring> Print the string on the screen.

CLEAR Clears the screen.

NFS <fileserver-command> Executes file-server or local command (e.g., MOUNT or SET DRIVE).

BOXCURSOR Turns on the M&R 80-column video Sup'R'Terminal card's box-like cursor.

LOWERCASE Sets lower case mode (^A^A^V) on the M&R card.

NOECHO <command> Executes <command> but doesn't print it.

ASK <command> Asks whether to execute <command>.

ASK=Y <command> Same as ASK, except that a carriage-return (only) response means yes.

ASK=N <command> Same as ASK, except that a carriage-return (only) response means no.

ASK ... PROMPT text The PROMPT option on ASK allows you to use your own prompt text. The prompt is NOT quoted and consists of whatever follows the word "PROMPT".

CONTINUE Asks if it is ok to continue. A reply of no means QUIT.

CONTINUE=Y Same as CONTINUE, except that a carriage-return (only) response means yes. The prompt on screen will be

CONTINUE (Y/N)?

and any typed response will replace the default Y.

CONTINUE=N Same as CONTINUE, except that a carriage-return (only) response means no.

SET USER id Sets an NFS lock with the id specified. The convention is that "id" are your initials and the lock name generated is USER.id. This allows a user to issue the NFS command "SHOW ALL LOCKS" to determine the initials of the users currently using the network.

SHOW FORMDATE Displays the date (and time if there is a Nestar Clock card in the NFS) on the screen. The format is:

Day dd-Mon-Year hh:mm:ss

NEWS Displays general system information (contents of PASCAL:NEWS.TEXT).

CHAIN <program-name> Passes control to the program named. Note that control is never returned to the startup program so commands following this command are not processed.

SET DATE Sets the Pascal system date to the

START

current date (from the NFS), as if the user had used the D(ate command in the Pascal Filer.

QUIT Exits the NESTAR.START program.

Example

```
; this is abcl:startcmd.text
PRINT
PRINT  ---->  THIS IS /MAIN/USERS/ABC/1  <----
PRINT
continue=y
set date
noecho boxcursor
noecho lowercase
noecho set user abc
noecho nfs set drive 3,6,7,8 virtual
nfs SET DIR /MAIN/USERS, PRVPW=xxx
nfs MOUNT ABC/TEMP,D9,RW,SHR
PRINT
ask=n news prompt display current news
```

This list of program commands begins with a comment ignored by the startup program ("this is startcmd.text"). When the volume is booted, "THIS IS /MAIN/USERS/ABC/1" is printed on the user-station screen. The user is then asked whether the startup program should continue or not. Typing "n" <return> quits the program and the Pascal command line comes up. Typing <return> causes the startup to continue.

The program then sets the Pascal system date to the current NFS date. The screen's box-like cursor and the lowercase options are turned on if an MnR Superterm card is being used. No indication of these two actions is given, because "noecho" is specified.

A user id is then set in the system and drives 3, 6, 7, and 8 are set to virtual (Section 7.2 discusses real and virtual drives in the Pascal environment). A default partial pathname is set, as well as a private password that may be required to access directories or virtual volumes

The virtual volume /MAIN/USERS/ABC/TEMP is mounted on drive 9 for shared, read/write use.

Finally, the user is prompted to display the system's news. Pressing <return>, however, is equivalent here to "no".

8.2.3 FILEINFO Utility Program

FILEINFO is a user-station utility program that runs in the Pascal environment and displays various information about files on an NFS disk. The file may be any type (directory, binary file, virtual DOS or Pascal disk, etc.). A versions of FILEINFO is also available for the BASIC environment (Section 8.3.3). Information provided by FILEINFO includes:

File name

Access rights (public, group, and private) and presence of passwords

Dates and times of creation, last access, last modification, and last backup

Total size in sectors

Virtual disk type (P, D, etc.)

Volume number (for DOS disks)

Useable size (in 512-byte blocks for Pascal or 256-byte sectors for DOS. This differs from the total size in sectors because there is one sector of overhead used by the NFS.)

For binary (T=B) files: the load address and size in bytes

For DOS (T=D) files: the simulated number of sectors per track and tracks per disk

8.2.3.1 How to Run FILEINFO

FILEINFO is distributed as part of the standard Cluster/One Model A File Server system. Ask your System Manager for the identification of the volumes on which this program resides at your installation. We assume here that the program is found on "PASCAL".

First, X(ecute PASCAL:FILEINFO from the Pascal command level. The Nestar logo will appear, followed by the prompt

Enter F(file info), NFS cmd, or Q(uit)

Enter "F" <return> to access file information. Type "N" <return> to enter a submode in which it is possible to execute NFS commands. Typing "F" causes a second prompt to

FILEINFO

appear:

Enter pathname of file (RETURN to exit)

You may use the full pathname from the root or any path from the current directory. You can display file information only for files to which you have read access. Any necessary passwords must be supplied with the pathname or specified by use of the SET command.

Suppose that the file to be displayed has the NFS pathname /MAIN/USERS/ABC/1. A private password, GOLD, has been set, and the default directory is /MAIN. Public and group access rights are restricted to Read. Typing in

USERS/ABC/1

will cause a display in the following format to appear:

```
File name: 1
This is a Pascal virtual disk
PUBACC=R
GRPACC=R
PRVACC=RWE,PW

Created           07 Mar 1981  03:39:56
Last access       02 Apr 1981  05:45:34
Last written      02 Apr 1981  11:21:43
Last backup       15 Mar 1981  10:33:55
Total size: 1601 256-byte sectors
Virtual disk type: P
Volume number: 1
Usable size: 800 512-byte blocks
```

Four dates are stored with each NFS file (including directories, binary files, pascal and DOS virtual disks, etc.). If a clock/calendar is installed in the system, the various date/time fields are updated as follows:

<u>Date</u>	<u>When Updated</u>
CREATED	When the file is first created with the CREATE command
LAST ACCESS	The first time daily that a file is mounted or a directory is used.
LAST WRITTEN	The first time after each mount that a file is written into. If a file is mounted for RW and never actually written into, this date will not be changed. For directories, this date is

changed any time an entry is changed by CREATE, DELETE, or RENAME.

LAST BACKUP

When the FILTRAN offline utility has been used to make a copy of all or part of a disk, and has been instructed to update the LAST BACKUP dates.

TDELETE

8.2.4 TDELETE Utility Program

TDELETE is a user-station utility program that runs in the Pascal environment and can be used to delete entire subtrees containing many files. (Remember that the NFS command DELETE will only delete a single file, and will delete a directory only if it is empty.)

To run TDELETE, ask the System Manager for the name of the volumes on which it resides. Assuming the name of the volume is PASCAL, boot a Pascal disk and from the command line X(ecute PASCAL:TDELETE. After the greeting banner, the program will display the command prompt

D(elete), N(FS command), Q(uit)?

Respond with N to enter a submode in which you can issue NFS commands. This is especially useful for using the LIST command to examine the directories to be manipulated. When in NFS mode, you may respond Q to return to the TDELETE command prompt.

Respond with D to delete a subtree. It will ask

ENTER PATH NAME OF SUBTREE TO DELETE:

After you enter the name of the directory whose files you wish to delete, it will list all the files in that directory (and all subdirectories) and then ask for confirmation:

DELETE THEM?

If you respond Y, then all the listed files (except the outer directory) will be deleted one at a time. The NFS commands being processed will be listed on the screen.

When all the entries in the directory have been deleted, it will ask whether you want the directory itself to be deleted:

DELETE <PATHNAME>?

where <pathname> is the original pathname that you supplied. If you respond Y, the directory will be deleted. In either case you will be returned to the TDELETE command prompt.

To exit the program respond with "Q" to the command prompt.

8.2.5 NETLINK Program

(This NETLINK documentation was originally distributed as NESTAR Technical Note #5.)

NETLINK is a utility program used to permit a user station which has booted Apple Pascal locally from Apple minidisks to access the Cluster/One Model A network.

Apple Pascal, as distributed by Apple Computer Inc., can access data stored on (real) minidisks. To permit it to access data stored on the Nestar File Server disks, certain modifications are made to the Pascal system as it is read into a user station by the BOOT program (when booting Pascal from the Nestar File Server). The NETLINK program makes these same modifications, but does so after Apple Pascal has been booted from a (local) minidisk.

Users who choose to regularly use their user stations in local mode, and only rarely utilize the network can use this program. Since the Apple Pascal system requires that you boot from slot 6, use of NETLINK avoids the need to move controller cards around. Simply leave the Apple in its normal configuration, and put the Nestar network interface card in slot 1, 2 or 3.

8.2.5.1 How to Run NETLINK

1. To execute NETLINK, and access the network, you must first have a standard Apple minidisk with two programs on it:

NETLINK.CODE and
CMD.CODE

These programs are distributed to your installation as part of the disk volume containing /PASCAL/LIB, and should have been copied to the user accessible volume by your System Manager. He can tell you how to use the Filer to copy them onto a real minidisk.

2. You should now boot your Apple into Pascal, using a minidisk controller in slot 6. The Nestar network interface card should be placed into slot 1, 2 or 3. The Pascal system will NOT boot if the Nestar network interface card is placed in slot 4, 5, or 7.
3. Now X(ecute the program NETLINK from the minidisk. The program will display the message:

PASCAL NOW LINKED TO CLUSTER/ONE NETWORK

NETLINK

and exit to the command level of Pascal.

NETLINK makes a test to see that a compatible version of the Pascal system is running before making any changes. If it does not support or recognize the version that is running, it will display the message:

THIS VERSION AND FLAVOR OF APPLE PASCAL
NOT SUPPORTED.

No linking has been performed.

and exit without making any changes.

NOTE: The only version of Apple Pascal currently supported by NETLINK is Apple Pascal Release 1.1.

4. You should now X(ecute the CMD program, and use it to set up any virtual drives, mounts, etc. you wish. Do not change drive 4 (the boot volume) from real to virtual under any circumstances!

For example, when you issue the command

SHOW DRIVES

you will see that initially all of the disk drives are considered REAL.

You may first issue the command

SET DRIVE 11,12 VIRTUAL

for example. Next you can MOUNT virtual disks on D11 and D12, and access them from Pascal, such as:

MOUNT /MAIN/PASCAL/LIB,D11

Note that drives 9,10,11 and 12 can be made virtual even if a real Apple minidisk controller is installed. The system will then access the corresponding virtual drives rather than the physical ones in slots 4 and 5. You can switch back as well (e.g. by executing CMD and entering SET DRIVE 11 REAL). As mentioned earlier, you must NOT set drive 4 to virtual, because it is the boot drive.

8.2.6 DIRLIST Utility Program

DIRLIST is a utility program operating in the Pascal environment that is used to save information provided by the LIST command (Chapter 5: LIST) for later use or reference. Note that the LIST command gives information about NFS files, not about files within a Pascal or DOS virtual disk.

To run DIRLIST, ask the System Manager for the name of a volume on which it resides. Assuming that the volume is named Pascal:, X(ecute PASCAL:DIRLIST from the Pascal command level. Following the greeting banner, a prompt will appear:

ENTER DIRECTORY PATHNAME TO BE LISTED (Q to quit)

Entering a pathname, such as /MAIN/USERS/ABC, will produce two further prompts, each reflecting LIST options:

NESTED LISTING? (Y/N)

VERBOSE LISTING? (Y/N)?

Choose the appropriate options and DIRLIST will prompt for the name of a file to write the information into:

FILENAME TO WRITE DIRECTORY LISTING TO? (<CR> FOR CONSOLE)

Enter a Pascal volume name and filename, not a Network pathname. The volume must be mounted read/write. Pressing just <return> will write the contents of the specified directory to the console (that is, will cause DIRLIST to execute the LIST command). Assuming that /MAIN/USERS/ABC has been typed in, with Y (for NESTED), Y(for VERBOSE), and <return>, DIRLIST will display:

Command: LIST /MAIN/USERS/ABC,NESTED,VERBOSE
buffering..
writing...

```

/MAIN/USERS/ABC,T=P,SIZE=3S
  AFILE,T=P,SIZE=801S
  BFILE,T=P,SIZE=1601S
  MAIL,T=P,SIZE=3S
    04060911.DEC,T=P,SIZE=3S

```

0,OK

0 lines (Mem available=15353 words)

ENTER DIRECTORY PATHNAME TO BE LISTED (Q to quit)

DIRLIST

/MAIN/USERS/ABC, then, is a directory containing the names of the two virtual Pascal disks AFILE and BFILE and the subdirectory MAIL, which lists one piece of mail in a file 3 sectors long.

Answering the prompt

"FILENAME TO WRITE...?"

with A:TEMP instead of <return> , where "A" is the Pascal volume name for AFILE, will write the listing above into a file named TEMP on the virtual volume "A".

DIRLIST then prompts for the next desired listing.

8.3 DOS User Station Programs

8.3.1 SETPROFILE Utility Program

SETPROFILE is a Nestar utility program run by the System Manager at a user station. The program creates a network file named /MAIN/SYSTEM/PROFILE/\$nn that contains information used by the NESTAR BOOT program, including the name of the volume to be booted at the user station when the power is turned on.

The user can specify to the System Manager whether the volume should be mounted automatically or only prompted for, and what the access rights to the volume are to be. If the volume is mounted automatically, any "hello" program on it will then be run.

SETPROFILE is explained in more detail in Section 5.8 of the System Manager's Manual. "Hello" programs are explained in the APPLE DOS Manuals.m1 4

DISK COPY

8.3.2 NESTAR DISK COPY Utility Program

NESTAR DISK COPY allows the user to copy the contents of an entire real or virtual disk (13 or 16 sectors), to another real or virtual disk, using any combination of Apple drives and Network files. NESTAR DISK COPY will copy BASIC or Pascal disks using 13 or 16 sectors per track formatting. DISK COPY cannot be used to copy VDOS CREATED disks or Pascal disks of variable size. NESTAR DISK COPY relies on Read-Write-Track-Sector (RWTS), the Disk Operating System's routine responsible for reading and writing individual sectors and for formatting all 35 tracks of a new disk.

NESTAR DISK COPY is a 6502 machine code program.

8.3.2.1 How to Run NESTAR DISK COPY

Ask the System Manager for the name of a volume on which DISK COPY resides.

Boot a BASICs disk that has the same sector per track formatting (13 or 16) as the intended original disk to be copied. If you intend to copy a 13 sector formatted disk, you must boot DOS 3.2 or 3.2.1 system. For copying a 16 sector disk, either a 16 sector BASIC or Pascal disk, be sure to boot a DOS 3.3 disk. Do not attempt to copy between disks of different sizes.

If you intend to use a single Apple drive it is suggested that you next type:

```
MAXFILES 1
```

in order to increase available RAM and speed of execution.

If you are going to use any Nestar Cluster/One virtual disks in the copy procedure (including the disk on which DISK COPY resides) then ensure that they are currently mounted on the desired drives.

Note that the Nestar File Server supports virtual drive numbers 1 through 255, so you may have more than just drives 1 and 2 available for multiple disk copies.

Also, be sure that NFS volumes are mounted RW if you want to copy to them.

You can now enter

BRUN NESTAR DISK COPY [,Sn,Dd]

After the welcome banner, the program displays the following information:

```
NESTAR DISK COPY 1.1    DOS/16 <-- 13 or 16 sectors
                        <-- DOS (vtoc) or Pascal (no vtoc)

ORIGINAL                COPY
(REAL)                  (VIRTUAL) <-- REAL=Apple, VIRTUAL=NFS

SLOT=4                  SLOT=6    <-- numbers entered by user
DRIVE=1                 DRIVE=99
VOLUME=1                VOLUME=123

READING                 <-- reading or writing

TRACK=11 SECTOR=0F      <-- track and sector currently
                        being copied (in hex)

PRESS RETURN TO CONTINUE <-- waiting for user to insert
                        original or copy disk
```

8.3.3 FILEINFO Utility Program

FILEINFO is a user-station utility program that runs in the DOS environment and prints various information about files on an NFS disk. The file may be any type (directory, binary file, virtual DOS or Pascal disk, etc.). Versions of FILEINFO are also available for the Pascal environment (Section 8.2.3). Information provided by FILEINFO includes:

File name

Access rights (public, group, and private) and presence of passwords

Dates and times of creation, last access, last modification, and last backup

Total size in sectors

Virtual disk type (P, D, etc.)

Volume number (for DOS disks)

Useable size (in 512-byte blocks for Pascal or 256-byte sectors for DOS. This differs from the total size in sectors because there is one sector of overhead used by the NFS.)

For binary (T=B) files: the load address and size in bytes

For DOS (T=D) files: the simulated number of sectors per track and tracks per disk

8.3.3.1 How to Run FILEINFO

FILEINFO is distributed as part of the standard Cluster/One Model A File Server system. Ask your System Manager for the identification of the volumes on which this program resides at your installation.

Boot a DOS volume that contains FILEINFO, type PR#n to enable the network card in slot n, and RUN FILEINFO. After the welcome banner, the prompt will appear:

ENTER VOLUME NAME (OR PRESS ESC KEY)

You may use the full pathname from the root or any path from the current directory set on your boot volume. You can display file information only for files to which you have

read access. Any necessary passwords can be supplied with the pathname.

Suppose that the file to be displayed has the NFS pathname /MAIN/USERS/ABC/1. A private password, GOLD, has been set, and the default directory is /MAIN. Public and group access rights are restricted to Read. Typing in USERS/ABC/1 will cause a display in the following format to appear:

----- /MAIN/USERS/ABC/1 -----

FILE TYPE: P (PASCAL)

PUBACC: R

GRPACC: R

PRVACC: RWE

CREATION: 3/ 7/81 3:39:56

LAST ACCESS: 4/13/81 5:45:34

LAST WRITTEN: 4/13/81 11:21:43

LAST BACKUP: 3/16/81 10:33:55

TOTAL SIZE: 1601 256-BYTE SECTORS

VIRTUAL TYPE: P

VOLUME NO. : 1 (INITIALIZED)

USABLE SIZE: 800 BLOCKS

ENTER VOLUME NAME (OR PRESS ESC KEY):

Four dates are stored with each NFS file (including directories, binary files, virtual disks, etc.). If a clock/calendar is installed in the system, the various date/time fields are updated as follows:

<u>Date</u>	<u>When Updated</u>
CREATION	When the file is first created with the CREATE command
LAST ACCESSED	The first time daily that a file is mounted or a directory is used.
LAST MODIFIED	The first time after each mount that a file is written into. If a file is mounted for RW and never actually written into, this date will not be changed. For directories, this date is changed any time an entry is changed by CREATE, DELETE, or RENAME.
LAST BACKUP	When the FILTRAN offline utility has been used to make a copy of all or part

FILEINFO

of a disk, and has been instructed to
update the LAST BACKUP dates.

8.3.4 FIDMORE

FIDMORE allows a user to load and modify a copy of Apple DOS 3.3 FID (a file transfer utility) to accept drive numbers from 1 to 9 instead of just 1 and 2. FIDMORE is a binary (machine language) program.

8.3.4.1 How to Run FIDMORE

Ask the System Manager for the name of a DOS volume upon which FIDMORE resides. Boot the volume and enter PR#6 to enable the network card in slot 6. Mount a volume containing FID (which is not a Nestar-distributed program) and BRUN FIDMORE.

After a welcome banner, the prompt will appear:

```
INPUT DRIVE NUMBER OF MOUNTED VOLUME CONTAINING FID
VERSION M:
```

Enter the drive number of the mounted volume containing FID.
The message

```
LOADING FID
```

will appear, followed by

```
FID WILL NOW ACCEPT DRIVE NUMBERS 1 to 9
```

After a brief pause, FID will come up. The version now in RAM will have been modified to accept drives 1 through 9. FID on the mounted disk remains unchanged.

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...

Appendix A

Error and Information Messages

When the file-server receives a file server command from a user-station, it executes the command, if possible, and, in any case, returns two variables:

a return code--an integer between 0 and 140

a return message, such as OK, ILLEGAL COMMAND, or TYPE=PASCAL

Chapters 6 (DOS) and 7 (PASCAL) describe how these variables are displayed in direct mode under DOS or by the Pascal CMD program and how to access them within a BASIC or Pascal program.

Messages are of several types:

information, such as OK, IN USE, TYPE=BINARY

user errors such as syntax errors or attempting to access a Network file without the necessary access rights

system errors such as memory full or disk full; these should be reported to the system manager.

In this list, messages are classified as:

general syntax

pertaining to a specific command

system errors

file system errors.

Specific hardware error codes and descriptions are listed in Appendix A of the System Manager's Manual.

ERRORS

General Syntax (0-19)

0,OK

(Not an error.) The command was syntactically correct and was executed without error.

1,ILLEGAL COMMAND

The command verb cannot be recognized, or cannot be executed from this user station operating system environment. The command verb is the first word of the command, and must be separated from the rest of the command with one or more blanks.

2,NAME PARAMETER REQUIRED

The first parameter of the command must be a non-null pathname.

3,UNRECOGNIZED PARAMETER

A keyword parameter was not recognized. Check the spelling carefully, and make sure that it is properly separated from the previous and following items with commas.

4,ILLEGAL DRIVE NUMBER

A drive number must be specified with "D" followed by an integer in the range 1 to 255. Example: "D4".

5,ILLEGAL STATION NUMBER

A station number must be specified as a two-digit hexadecimal number preceded with \$, in the range \$1 to \$FE. Example: "\$2D".

7,PARAMETER APPEARS TWICE

A keyword parameter appears more than once in a single command. There can only be one occurrence of a keyword parameter, even if multiple occurrences have the same value.

8,ILLEGAL TYPE PARAMETER

The "T=" value is not one of the legal volume types. It must be one of B (binary), D (DOS), P (Pascal), Y (directory), S (system). Example: "T=P".

9, COMMAND TOO LONG

An NFS command must be no longer than 80 characters. In unusual cases where this is a problem, long commands can often be shortened by using a default directory, or by assigning access rights in a subsequent PROTECT command.

10, PARAMETER TOO LONG

The new name given as the second parameter of the RENAME command must be no more than 15 characters long. Remember that RENAME changes only the last name in the sequence of names which constitute a pathname. If you wish to change the name of a directory, give the pathname of the directory itself as the first parameter of RENAME.

11, ILLEGAL PROTECTION PARAMETER

The value of the PUBACC, GRPACC, or PRVACC protection item is incorrect. The value must be some combination of the letters RWECD (read, write, erase, create, and delete), or may be null. For example, "GRPACC=RW" or "PUBACC="

12, NO STATION CMD; USE \$NN PREFIX

The "station" command has been removed. Use "\$nn <command>" where nn is the station number on whose behalf you wish to issue the command. (NFS console only.)

13, ILLEGAL NUMBER (NEGATIVE OR TOO BIG)

The number was less than 0 or bigger than 32767. Number may be specified in decimal, or in hexadecimal preceded by \$. Examples: "42", "\$1A3".

19, HELP FILE NOT FOUND

There is no HELP file for the word used as an operand of the HELP command. Use the HELP command with any parameters to see what the possible HELP parameters are. (The system manager has the option of removing all HELP files to save space.)

Create (20-25)

ERRORS

20,TYPE PARAMETER REQUIRED (T=)

The type parameter was missing. Example: "T=D".

21,SIZE OUT OF RANGE (<0B OR >32752B)

The value of the SIZE parameter was negative or greater than 32752 blocks. Each block is 512 bytes. Other units that can be specified are C (characters), S (256-byte sectors), and K (1024 bytes). If the units are omitted, B (512-byte blocks) is assumed. Examples: "SIZE=280", "SIZE=2000S".

23,SIZE PARAMETER REQUIRED WITH T=B

The SIZE parameter must be specified to create a binary volume. The size may be an overestimate of the size of the data to be BSAVED into the volume, but it must not be less.

24,SIZE OUT OF RANGE (<1C OR >48K)

The limits on the size of a binary (T=B) file are between 1C (one byte) and 48K (49152 bytes) regardless of the units used in the specification.

25,SECTORS PARAMETER REQUIRED WITH T=D

When creating a DOS (T=D) volume, you must specify the number of sectors per track. (Use SECTORS=13 for DOS 3.2.1 and earlier volumes, or SECTORS=16 for DOS 3.3 volumes. For information about varying the size and number of sectors per track, see the section on VDOS CREATE in the User's Manual.)

Delete (26-29)

29,CURRENTLY MOUNTED OR DEFAULT DIR

You cannot delete a volume which is currently mounted by any station, or which is the default directory (see the SET DIR command) of any station.

Mount (30-34)

30,DRIVE REQUIRED

You must specify what drive the volume is to be mounted

on. Example: "D4".

31, IN USE

The volume you asked to mount is in use by another station. You are therefore denied exclusive use of that volume.

32, IN EXC USE

The volume you asked to mount is in exclusive use by another station. You are therefore denied any use of the volume.

34, RW NOT ALLOWED ON DIRECTORIES

You are not allowed to mount a directory (T=Y) volume for write access.

Unmount (35-39)

35, DRIVE OR "ALL" REQUIRED

You must specify a drive number or "ALL" on the UNMOUNT command. Examples: "UNMOUNT D1", "UNMOUNT ALL".

36, NOT CURRENTLY MOUNTED

The drive number specified in the UNMOUNT command does not have any virtual volume mounted on it by the NFS.

Lock (40-44)

40, ILLEGAL LOCK NAME

A lock name must be 1 to 15 characters long.

41, IN USE

The lock name specified is in use by another station. You are therefore denied exclusive use of the lock.

42, IN EXC USE

The lock name specified is in exclusive use by another station. You are therefore denied any use of the lock.

ERRORS

Unlock (45-49)

45,NOT HELD

The lock name specified is not currently held by your station.

Bsave/Bload/Brun (50-55)

51,NOT BINARY

The pathname identifies a volume which is not binary (T=B). Only binary volumes can be used for BSAVE, BLOAD, or BRUN commands.

52,NOT PREVIOUSLY SAVED

The binary volume was created but has never had any data written into it using the BSAVE command. It can not not be used with BLOAD or BRUN until it has been BSAVED into.

53,FROM PARAMETER REQUIRED

For BSAVE, the FROM parameter, which specifies the memory area from which data is to be taken, must be provided. Example: "FROM=\$1000".

54,SIZE PARAMETER REQUIRED

For BSAVE, you must provide the SIZE parameter on the BSAVE command to indicate how much data is to be saved. This value must be less than or equal to the size of the volume specified on the original CREATE command. Example: "SIZE=256.C".

55,FILE WAS THERE; REP NOT SPECIFIED

Data has already been saved in the binary volume. To replace that data, you must specify "REPLACE" as a parameter in the BSAVE command.

Set/List (56-58)

57,FILE IS NOT A DIRECTORY

The pathname given in a SET DIR command identifies a

volume which is not a directory (T=Y).

58,DATE MAINTAINED AUTOMATICALLY

The SET DATE command cannot be used when the clock/calendar option of the file server station is installed. The date and time are maintained by the clock/calendar.

Show (59-69)

59,TYPE=SYSTEM

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a system volume (T=S).

60,SHOW OPTION NOT RECOGNIZED

The operand of a "SHOW" command was not recognized. Check the spelling and make sure there are no extraneous items in the command.

61,DRIVE REQUIRED

For the "SHOW TYPE" command you must specify the drive number on which the volume is mounted. Example: "SHOW TYPE D12".

62,NOT CURRENTLY MOUNTED

The drive number specified has no volume mounted on it by the NFS.

64,NO DEFAULT DIRECTORY SET

(Not necessarily an error.) This response to the "SHOW DIR" command indicates that there is no default directory established for this station by the NFS.

65,TYPE=PASCAL

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a Pascal volume (T=P).

66,TYPE=DOS

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a DOS volume (T=D).

67,TYPE=BINARY

ERRORS

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a binary volume (T=B).

68,TYPE=DIRECTORY

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a directory (T=Y).

69,UNKNOWN TYPE

The volume mounted on the drive for which you did a "SHOW TYPE" is not a standard volume type.

70,yyymmddhhmmssw

(Not an error) Returns date and time information in encoded form. For example, the NFS command `TIMESTAMP` returns the current year, month, day, hour, minute, second, and day of the week in the form 70,8104301859381 (that is, 6:59 P.M., Sunday, April 30, 1981).

71,--t--tt--t-t----

(Not an error) Returns types of virtual disks currently mounted on drives 1-16 (Y, P, D, B). The system responds to the NFS command `SHOW TYPES` using this format.

Local Commands (90-92)

90,DRIVE NUMBER(S) REQUIRED

A drive number, or list of drive numbers, is required for this command. Example: "SET DRIVES 4,5 VIRTUAL".

91,DRIVE NUMBER CAN NOT BE USED

There are 12 drives (also called "volumes") in Pascal, but only some of them can be made virtual with the SET DRIVE local command. In Pascal Version 1.0, drives 4,5,9,10,11, and 12 are valid. In Pascal Version 1.1, drives 3 through 12 are valid.

92,NO MORE CHANNEL SPACE (MAX=3)

In the Pascal environment, a "channel" is established for each different file server you have simultaneous access to. There is space for three channels, so you

ERRORS

may have drives virtual to no more than three file servers at the same time. There also must be a channel established for the default file server to which commands are sent. This error indicates that you have asked for more than three file server channels to be established.

System Errors (94-99)

94, NOT CONNECTED

This is a network error that indicates that the connection to another station has been terminated prematurely.

95, UNRECOGNIZABLE NETWORK RESPONSE

The last response message from file server did not contain a numeric error code, as expected. This may indicate incorrect logic in the use of NETUNIT procedures from within Pascal programs.

96, NO NETWORK CARD IN SLOT

The slot number specified does not have a Nestar network interface card installed, or the card is defective.

97, MEMORY FULL

The file server has no space left for tables needed to complete your request. See the System Manager's Manual for information about file server table space. This does NOT refer to memory space in the user station.

98, NETWORK ERROR

The network routines were unable to complete the transaction. The error may be transient; see the description of NETUNIT for more information. This error does not occur from most user-level programs, such as CMD.

99, USER ABORT

The network-abort key (CNTL-SHIFT-N) was pressed while a network transaction was queued up or in progress. The transaction was aborted.

ERRORS

NFS File Errors (101-134)

101,END OF FILE OR RECORD OUT OF RANGE

An attempt was made to read a block or sector of a volume which is outside the legal range for that volume.

102,FILE NOT FOUND

The volume specified by the pathname was not found, or one of the directories in the pathname was not found. Check the spelling of each filename in the pathname. If the name does not begin with a slash, check that the current default directory is the correct one.

103,BAD DELIMITER IN PATHNAME

A delimiter other than "/" or ":" was found in a pathname. Make sure that you have not omitted a comma separating the pathname from other operands in the command.

104,FILE NAME OR PASSWORD TOO BIG

A single filename (the part between slashes in a pathname), or password (the part after a colon in a pathname) is longer than the maximum of 15 characters.

105,NO ACCESS FOR READ TO DIRECTORY

You have been denied read access to a directory which is part of the specified pathname.

106,NON DIRECTORY FOUND IN PATHNAME

One of the names in the pathname (other than the last name) identifies a volume which is not a directory (T=Y). Only the last thing in a pathname can be other than a directory.

107,END OF PATHNAME IS A DIRECTORY

The pathname specifies a directory (T=Y) in a context where a non-directory volume is required.

109,ROOT DIR NOT SPECIFIED; NO DEFAULT

The specified pathname does not begin with a slash, and there is no default directory recorded for this station by the file server. If you wish to completely specify

the pathname, begin with a slash and the name or number of the disk unit. If you wish to use the current default directory, do not begin the pathname with a slash. You may use the "SHOW DIR" command to find out the current default directory, and the "SET DIR" command to establish one.

110,NO ACCESS FOR READ

You have been denied read access to the volume specified by the pathname, or to a directory along the path.

111,NO ACCESS FOR WRITE

You have been denied write access to the volume specified by the pathname.

112,NO ACCESS FOR APPEND (not yet implemented)

You have been denied append access to the volume specified by the pathname.

113,NO ACCESS FOR ERASE

You have been denied erase access to the volume specified by the pathname. Erase access is necessary to delete the file.

114,NO ACCESS FOR CREATE

You have been denied access to create or rename an entry in one of the directories specified in the pathname.

115,NO ACCESS FOR DELETE

You have been denied access to delete or rename an entry in one of the directories specified in the pathname.

116,CANT DELETE NON-EMPTY DIRECTORY

The pathname specified in a DELETE command identifies a directory, and that directory is not empty, that is, it still points to other volumes. Only empty directories can be deleted by a single file server DELETE command. To delete non-empty directories (i.e. subtrees) see the description of the TDELETE utility in the User's Manual.

117,MEMORY FULL

ERRORS

The file server has no space left for tables needed to complete your request. See the System Manager's Manual for information about file server table space. This does NOT refer to memory space in the user station.

118,DISK FULL

There is not enough contiguous space left on the disk unit to create the volume. The LIST command when used to display the root directory will give information about the space available on a disk unit. Example: "LIST /MAIN".

119,DIRECTORY FULL

There is not enough space left to create more entries in the directory, and the directory cannot be expanded. (In NFS version 1.1 and later, directories are automatically expanded as necessary to accomodate new entries.)

120,INTERNAL ERROR

An internal error has been detected by the file server. Additional information is written on the console of the file server. That information and the circumstances surrounding the error should be transmitted to Nestar for diagnosis. The file server should be restarted as soon as practical.

121,UNINITIALIZED DISK

The disk unit (hard disk or floppy disk) has not been initialized. All disks must be formatted and initialized using the Nestar FORMAT program; see the System Manager's Manual for details.

122,WRONG SOFTWARE VERSION

The disk format is not compatible with version of the file server currently running. (This error cannot occur with any file servers so far released.)

123,FILE ALREADY EXISTS

The file you have asked to create, or the new name used in a RENAME command, already exists in the directory.

124,DISK I/O ERROR, SUBCODES x,y

A hardware I/O error was detected. The details of the

error are described in the subcodes; see the table of I/O errors for more information.

125,VERIFY FAILED (BAD MEMORY)

All disk write operations are verified by reading back the recorded data and comparing it to the data stored in memory. This error indicates that the disk data did not compare correctly. It sometimes indicates a memory error in the file server, and not a disk error.

126,DISKETTE WAS CHANGED

The door to the floppy disk unit was opened. The file server assumes that the diskette was changed, and does not allow operations to proceed. See the description of the DISK CHANGE command in the System Manager's Manual for a discussion of safely changing floppy diskettes which are in use.

127,BAD DISKETTE FORMAT

The floppy diskette is not correctly formatted. All diskettes must have been formatted by the Nestar FORMAT program; see the System Manager's Manual for details.

128,BAD UNIT NUMBER IN PATHNAME

The first item after the initial slash in a pathname is a number, but it is non in the legal range for unit numbers (1 to 6).

129,DISKETTE WRITE PROTECTED

An attempt was made to write to a floppy diskette which is write protected. Remember that the convention for 8" diskettes is opposite to that for mini-diskettes: the hole must be covered to allow writing. See the System Manager's Manual for more details.

130,NEED PRIVATE PASSWORD FOR PROTECT

In order to execute the PROTECT command for any volume, the private password must be specified in the pathname or as a default private password. The password must be specified whenever the volume has a non-null private password, regardless of the access rights currently assigned to the volume.

131,UNIT NAME NOT FOUND

The name after the initial slash of a pathane is not

ERRORS

the name of a disk unit currently recognized by the file server. If a floppy diskette has just been inserted into a drive, use the "SHOW VOLS" command to have the file server search for all new disk units and show you the results. After that, all displayed names will be recognized as disk unit names. You can also use the unit number in place of a name, if you wish.

133,DISK NOT READY

The disk unit specified is not ready. For a floppy disk, it may indicate that no diskette has been inserted. For hard disks, the three minute warmup period after initial power up may not have elapsed. It may also indicate a controller or disk drive failure.

134,DISK CONTROLLER NOT PRESENT

There is no controller installed for the disk unit, or the controller is defective. The controller for floppy disk units 1 and 2 goes in slot 4 of the NFS, and controllers for hard disk units 5 and 6 go in slots 2 and 3 of the NFS, respectively.

Appendix B

The BOOT Program

B.1 Description

The BOOT program is a standard part of the Cluster/One Model A File Server, and provides a means for users to load a local operating system such as Apple DOS 3.2, Apple DOS 3.3, Apple Pascal 1.0, or Apple Pascal 1.1. It is automatically loaded and run when an Apple with the autostart ROM and a Nestar network interface card is powered up. If an Apple disk controller is in a higher slot than the network card, however, the Apple will boot from the local disk.

The BOOT program is a machine language program which executes in the memory area \$800 to \$1FFF. It has been provided as a BSAVED virtual Network file named /MAIN/BOOT. This section describes the action of BOOT as distributed by Nestar. You should consult your System Manager to see if any local modifications have been made to the version on your Network File Server.

BOOT is brought into a user station's memory and executed whenever the Apple Autostart ROM (present on Apple II Plus, Apple II with Language Card or Integer machines which have upgraded by installing the ROM) recognizes a power on condition. On machines which do not have the Autostart ROM, you must explicitly enter

Cn00G (n is the slot in which
 a Nestar network interface
 card is installed)

in response to the Apple II Monitor prompt ("*"). Entering PR#6 or 6ctrlP will NOT cause the BOOT program to be run. The Network interface card differs from Apple Disk II controllers in this respect; they will boot whenever any character is printed to the slot they are plugged into.

When a user station is powered up BOOT will automatically unmount all disks in use by that station, free any locks held, and reset any default directory or passwords. This also happens when the OFF file server command is issued. It is strongly recommended that this be done by every user who is finished using an Apple on the Model A system. This gives a clear indication to others that the station is not in use, and also frees up resources other stations may need.

BOOT

B.2 Operation of BOOT

BOOT will display on the Apple screen the Nestar logo, followed by:

CLUSTER/ONE MODEL A NETWORK BOOT 1.1

STATION ADDRESS \$NN
SLOT N SERVER \$FE

VOLUME TO BOOT: /MAIN/

where NN is your station address (in hex, from 00 to 7F), and N and \$FE are the SLOT and Server numbers that communications are directed to. You should place a sticker with the station number on each user station, and you should observe that the number shown on the screen agrees. If not, please consult your System Manager.

NOTE: The boot program recognizes a large collection of Apple-compatible serial or communications cards installed in slot 3, and directs its I/O to them automatically. Whenever I/O is being directed to the slot 3 device, the message

(I/O BEING DIRECTED TO SLOT 3 DEVICE)

is displayed on the standard Apple video output as an indication to the user.

A user may disable the automatic recognition of a slot 3 terminal-like card by pressing the space bar down between the time that the BOOT program is downloading and begins running. If this is done, the standard Apple video output and keyboard input will be used regardless of a card in slot 3.

The use of a slot 3 card under DOS and the Model A is not especially convenient, since commands are routed to the Network File Server using PR#n (n is the slot in which the Nestar card is installed), while the slot 3 card only displays output when a PR#3 is performed.

The user may now enter the volume name of a DOS, PASCAL, or BINARY disk he wishes to load his system from. This name may be up to 40 characters long, including the prefix. A default prefix is displayed on the screen (shown above as /MAIN). The user may backspace and edit this prefix exactly as if he had typed it himself. BOOT will BRUN a binary file if the file type of the file supplied to the VOLUME TO BOOT? prompt is binary.

You should note that use of the Model A Network File Server does not in any way modify the minimum features necessary to load and execute DOS or Pascal. Any errors you would get from these standard systems will still be reflected in the Model A system. For example, you cannot boot a Pascal system into a machine which does not have a Language Card. DOS slave disks initialized on a 48K Apple will not run on a 32K Apple. However, the BOOT program does support virtual DOS disks of variable size: 4 to 32 sectors per track and 18 to 50 tracks per disk. Certain things are done for you automatically, as a convenience to the user, wherever feasible. These are discussed in the following paragraphs.

After you have entered the name of the volume you wish to boot, the BOOT program attempts to locate that volume, and issue a MOUNT command for drive 1. If successful, the type of the disk is checked, and DOS is distinguished from Pascal. The volume is mounted read/write, update unless the individual station profile or system default profile files have been changed for different access and usage. (discussed below in Section B.2.3).

B.2.1 Booting DOS Virtual Diskettes

For a DOS disk, the system reads the first sector, and determines whether this disk appears to be a DOS 3.2 (i.e. 13 sector disk) or a DOS 3.3 (i.e. 16 sector disk or VDOS CREATED variable sector disk), at which time it begins the bootstrap process in a fashion parallel to the action of booting a physical minidisk. The DOS is loaded, relocated if a master disk, and then started. The HELLO program, if any, will be run, as is standard.

As a convenience to the DOS user, BOOT checks to see if a Language Card is present, and makes an attempt to load the Language Card with whichever version of BASIC is not present in ROM. Unless changed by the installation, this is done by locating a Pascal volume, called /MAIN/PASCAL/LIB, and locating the files FPBAS.DATA or INTBAS.DATA. (These files should have been installed by your System Manager during installation.)

B.2.2 Booting Pascal Virtual Diskettes

In the case of booting a Pascal disk, a check is made to see that the Language Card is present before attempting to load Pascal. If present, the Pascal volume /MAIN/PASCAL/LIB is located, and mounted RO and SHR on drive 5 of the users station. The volume the user requested to boot is remounted on drive 4, for RW and UPD usage or for the usage specified in the station's profile in /MAIN/SYSTEM/PROFILE.

BOOT

The file SYSTEM.APPLE must be present on PASCAL/LIB or on the user's booted volume. If SYSTEM.APPLE is not found on either, an error message is issued.

Once SYSTEM.APPLE is loaded into the Language Card, the Pascal system is started. In the case of booting Apple Pascal Release 1.1, SYSTEM.PASCAL need not be present on the booted disk, and can be loaded from the library disk instead. When booting Apple Pascal 1.0, the booted disk MUST contain the file SYSTEM.PASCAL and SYSTEM.MISCINFO, with the appropriate GOTOXY installed for the type of terminal the user is using. BOOT checks the version numbers of SYSTEM.APPLE and SYSTEM.PASCAL, and will only accept Apple Pascal 1.0 or 1.1 systems. It also verifies that the versions of SYSTEM.APPLE and SYSTEM.PASCAL are at the same level.

In the typical Pascal system, Pascal looks for the three necessary files, SYSTEM.STARTUP, SYSTEM.LIBRARY, and SYSTEM.MISCINFO, on the user's booted volume. However, in the Cluster/One system, the System Manager may run the utility program ZAPPASCAL against SYSTEM.PASCAL to change the names of the three files searched for to #5:NESTAR.START, #5:NESTAR.LIB, and #5:NESTAR.MISC. These may reside on the drive 5 library volume, so that it is no longer necessary for all users to have the files on their boot volumes. See the description of the utility program ZAPPASCAL in the System Manager's Manual for more details.

Pascal will now come up and follow its normal procedures. It will recognize a serial or communications card in slot 3, and direct its subsequent input/output to that card if present.

B.2.3 Autobooting at the Individual Station

The BOOT program can be tailored to the needs of individual stations. Using the utility SETPROFILE, the System Manager can install a profile for any station. The profile will specify which volume is to be booted for the station, whether the boot is to be automatic, the Pascal and DOS access rights, and the name of the default library volume automatically mounted on drive 5. Individual users can install station profiles themselves using SETPROFILE if they have the access rights necessary to do so.

If no profile has been set for a station, BOOT will use the defaults set by the System Manager in the system \$00 profile.

If <esc> is pressed on the Apple keyboard while the BOOT

BOOT

program is downloading, the station dependent profile is ignored and the default profile is used.

1. All the information in this report is confidential and should be handled accordingly.

Appendix C

Changes to Cluster/One Model A in Release 1.1

<u>Section</u>	<u>Version 1.1</u>
1.4	Supports two hard disks, up to 66 megabytes shared disk capacity.
2.4ff 6.1ff	Supports Apple DOS 3.3
2.5	Notation "T=V" in directory listings is replaced by the proper type designator (e.g., D, B, P).
3.1.3	Directories are automatically expanded as necessary.
5.0 (MOUNT)	MOUNT allows a new usage mode, UPD, which permits one user RW access to a file, while others may mount the file RO,SHR at the same time.
5.0 (CREATE)	CREATE command can now be used to create and mount a file in one command.
5.0	New NFS commands: RENAME, TIMESTAMP, SHOW ALL LOCKS, SHOW ALL MOUNTS, SHOW TYPES
6.6	DOS 3.3 virtual diskettes can be of variable size and need not be INITed. (Use VDOS CREATE)
7.0ff	Supports Apple Pascal 1.1.
8.1	New Pascal utilities allow users to tailor a central Startup program for use at their individual stations (NESTAR.START, ZAPPASCAL), access and manipulate the file system in new ways (FILEINFO, TDELETE, DIRLIST), and access Network resources and facilities from within their local pascal programs (DATEUNIT, NETLINK); note especially extensive changes in NETUNIT. New program SETPROFILE allows the System Manager to create

CHANGES

a file of autoboot defaults for the individual station.

8.2.2

New DOS utility FIDMORE allows a user to load and modify a copy of Apple DOS 3.3 FID to accept drive numbers from 1 to 9. New program SETPROFILE allows the System Manager to create a file of autoboot defaults for the individual station.

Appendix B

BOOT program changes allow each station to have its own auto-boot defaults.

System Manager's Manual

Optional NFS Clock/Calendar

Appendix D

Reading the Station Number From a Program

D.1 Introduction

It may be useful to read the station number (1 to 127) from within a program. This can provide a means of writing one standard HELLO program which then MOUNTs, RUNs, etc. different programs or disks based on the station number. A similar strategy can be used in Pascal by embedding the station number reading and a series of special MOUNTs, etc. within a program called SYSTEM.STARTUP.

The current station number is held in memory on the Nestar bus interface card, and can be read from there. This area of memory is addressable only by carefully following the Apple conventions for bank switching.

The following three program excerpts demonstrate how to access the one byte address field from Assembler, Basic or Pascal.

All three examples assume the Nestar bus interface card is in slot 6. If some other slot is used, the appropriate address (\$Cn00, where n is the slot number) should be used in the second step of the programs shown.

D.2 Machine Language

```
GETADR  LDA $CFFF      ; Deselect all cards
        LDA $C600      ; Select bus card (assuming slot 6)
        LDA $C800      ; Get the address
        RTS            ; Return with it in A-register
```

D.3 BASIC

```
10 AD = PEEK(-12289) :REM Deselect all cards
20 AD = PEEK(-14848) :REM Select bus card (assuming slot 6)
30 AD = PEEK(-14336) :REM Now you have the address 0 to 127
```

READ STN #

D.4 Pascal

```
FUNCTION GETADDR:INTEGER;  
  
TYPE BYTE = PACKED ARRAY[0..0] OF 0..255;  
  
VAR TRIX: RECORD CASE BOOLEAN OF  
    FALSE: (ADR:INTEGER);  
    TRUE: (VAL:^BYTE)  
END;  
  
BEGIN  
    TRIX.ADR := -12289;  
    GETADDR := TRIX.VAL^[0]; (* Deselect ALL CARDS *)  
    TRIX.ADR := -14848;  
    GETADDR := TRIX.VAL^[0]; (* SELECT CARD IN SLOT 6 *)  
    TRIX.ADR := -14336;  
    GETADDR := TRIX.VAL^[0]; (* RETURN ADDRESS VALUE *)  
END;
```

D.5 Special Locations in the Network Interface Card

Assembly language programs running on a station may select the Nestar Network Interface Card by executing:

```
JSR $Cn19
```

This routine has the effect of deselecting any other card using the \$C800 address block, selecting the Nestar card, and putting \$Cn in \$07F8 (the active slot location.)

Once the Nestar communications card is selected, the Station number may be read from location \$C800.

Appendix E

Quick Reference List - File Server Commands

	See also <u>Page</u>
BLOAD pathname [,LOAD=number]	5-5
-loads binary virtual disk previously saved with BSAVE	
-loads at LOAD address in BLOAD, if present; if not, loads at LOAD address in corresponding BSAVE, if present; otherwise, loads at FROM address in BSAVE	
BRUN pathname [,LOAD=number]	5-5
-BLOADS a binary virtual disk, then executes it starting at first address loaded	
BSAVE pathname, FROM=number, size [,LOAD=number] [,REPLACE]	5-5
-saves core image on disk of specified size, starting at FROM address	
CREATE pathname, type [,size] [,sectors] [,protection] [,drive] [,usage]	5-9
-creates and optionally mounts a new virtual disk	
DELETE pathname	5-14
-deletes a virtual disk from system	
HELP [command name] or ? [command name]	5-15
-displays list of commands or syntax of specified command	
LIST [pathname] [,VERBOSE] [,NESTED]	5-16
-lists entries in default or specified directory	
LOCK lockname [,usage]	5-19
-locks "lockname" for exclusive (EXC) or shared (SHR) use	
-Default: EXC	

COMMAND LIST

	See also <u>Page</u>
MOUNT pathname, drive [,usage] -establishes correspondence between a virtual disk and drive number d	5-21
OFF -initializes user station and runs /MAIN/BOOT	5-23
PROTECT [pathname] [protection] -modifies passwords or access rights for a virtual disk or directory	5-25
RENAME oldpathname, newname -renames specified NFS file	5-28
SET DIR pathname -sets default directory	5-29
SET [GRPPW=password] [PRVPW=password] -sets default group or private password	5-31
SHOW DATE -displays date as MM/DD/YY	5-33
SHOW DIR -displays name of current default directory	5-34
SHOW LOCK [lockname] -lists all stations using specified lockname	5-35
SHOW [ALL] LOCKS -displays all locks set at own station or held in file server	5-36
SHOW [ALL] MOUNTS [pathname] -gives information on virtual disks mounted on the file server	5-38

	See also <u>Page</u>
SHOW PROTECTION [pathname]	5-41
-displays group, private and public access rights for virtual disk or directory addressed by pathname	
SHOW STATION	5-42
-displays own station number	
SHOW TYPE Dd	5-43
-displays type of virtual disk mounted on drive d	
SHOW TYPES	5-44
-displays types of virtual disks mounted on drives 1-16	
SHOW VOLS	5-45
-displays names of system disk units currently operational	
TIMESTAMP	5-46
-returns encoded form of the current date and time	
UNLOCK lockname or UNLOCK ALL	5-47
-releases locked usage on a lockname or on all locknames currently held	
UNMOUNT Dd or UNMOUNT ALL	5-48
-cancels previous mount(s)	

Parameters

access rights: set of letters from RWECD (read, write, erase, create, and delete)	4-4
drive: Dd, where d=number (1-255 for DOS; 3-12 for Pascal)	6-1 7-2
filepath: name[:password] [/filepath]	3-3
lockname: name of lock (same restrictions as name)	4-9

COMMAND LIST

	See also <u>Page</u>
name: name of disk unit, directory, or virtual disk. (1-15 characters; no commas, control characters, slashes, colons, returns or unprintable characters; imbedded blanks ok; first character any ok character)	3-3
number: if decimal: between -32768 and 32767, inclusive if hex: between \$0 and \$FFFF, inclusive	3-3
password: a secret word used to gain access to protected NFS. Same restrictions as name.	4-4
pathname: /unitname/filepath or filepath (80 characters max. An initial slash indicates first field is a unit name; no initial slash indicates current default directory is to be prefixed to pathname.	3-3
protection: protect item [,protection]	4-4
protect item: ,GRPPW=password ,PRVPW=password ,PUBACC=access rights (Default: R) ,GRPACC=access rights (Default: PUBACC) ,PRVACC=access rights (Default: See 4.1.3)	4-4
sectors: SECTORS=13 for DOS 3.2.1; SECTORS=16 for DOS 3.3; variable for DOS 3.3 using VDOS CREATE (Sec. 6.6)	5-9 6-4
size: SIZE=number [.size-unit] (must be given for binary)	5-9
size-unit: C: character (1 byte) S: sector (256 bytes) B: block (512 bytes) blank: block (512 bytes) K: 1024 bytes	5-5 5-9 5-5
type: T=B (binary) T=D (DOS) T=P (Pascal) T=Y (directory)	5-9
unitname: 1, 2, or 5 [:password] or name [:password]	3-1
usage: EXC (exclusive), SHR (shared), or UPD (update) RO (read only) or RW (read/write)	4-9 5-21

Local CMD Commands

SHOW DRIVES [NOCHECK]	7-4
SHOW NFS	7-5
SHOW SLOTS	7-5
SET DRIVE[S] n[,n,...] REAL VIRTUAL [SLOT n] [STN \$nn]	7-5
SET NFS [SLOT n] [STN \$nn]	7-6
Other CMD Inputs	7-6
drive: 3-12	7-2
drive-status: R, REAL, V, or VIRTUAL	7-2
slot: default virtual slot	7-6

A

access rights: 4-1, 4-5, 4-7
 ALL: 5-36, 5-38
 Apple: 1-1, 1-4
 Applesoft: 6-2

B

BASIC: 2-1..2-14, 6-1..6-6
 binary: 5-3, 5-5..5-8
 BLOAD: 5-5..5-8, 5-22
 BOOT Program: 6-1, 7-2, B-1..B-3
 BRUN: 5-5..5-8, 5-22
 BSAVE: 5-5..5-8

C

Changes to Cluster/One Model A in Release 1.1: C-1
 CMD: 7-3..7-10, E-4
 CMDUNIT: 7-8
 commands: 5-1
 copying from a cassette tape: 2-12
 copying from a diskette: 2-13, C-1..C-4
 copyright: i
 CREATE: 5-9..5-12
 ctrl-d: 2-10, 6-2
 ctrl-n: 2-10, 6-3
 ctrl-shift-n: 7-3

D

DATE: 5-34, 5-46
 decimal numbers: E-3
 default access rights: 4-5
 default directory: 3-4, 5-3, 5-29..5-30, 5-32
 default parameters: 5-2
 default passwords: 4-6, 5-3, 5-29..5-30
 DELETE: 5-14
 directory: 2-2, 3-1..3-4
 default: 3-4, 5-3, 5-29..5-30, 5-34
 root: 3-1..3-2
 DIRLIST: 8-13
 disclaimer: i
 disk controller: 1-4, 7-2
 disk copy routine: 2-13, 8-16
 disk organization: 2-2, 3-1
 disk units: 3-1, 5-45
 DOS with the Cluster/One system: 6-1
 drive: 5-20, 5-39, 5-42, 6-1, 7-1..7-3, 7-4, 7-5

INDEX

E

error and information messages: A-1..A-4
EXC: 4-9, 5-11, 5-21
exclusive usage: 4-9, 5-11, 5-21

F

FIDMORE: 8-21
field: 5-2
FILEINFO: 8-7 (Pascal), 8-18 (DOS)
File Server: 1-1, 1-4
file server command: 5-1, 5-4, 7-7, E-1..E-4
 from Applesoft: 6-2
 from Integer BASIC: 6-3
 from Pascal: 7-7
 with DOS: 2-5, 6-2
filename: 3-5
filepath: 3-5
floppy disk: 1-2, 3-1
FROM: 5-5

G

group access rights: 4-2..4-5
GRPACC: 4-4..4-5, 5-10, 5-25..5-26
GRPPW: 4-4..4-5, 5-10, 5-25..5-26

H

hard disk: 1-2, 1-3
HELP: 5-15
How to Use This Manual: ii

I

I/O Devices: 1-3
IN#n: 2-11, 6-3, 6-4
INIT: 2-9, 6-4
initializing a DOS virtual diskette: 2-9, 6-4
initializing a Pascal virtual diskette: 7-3
Introduction: 1-1
issuing NFS commands within a BASIC program: 2-10..2-12, 6-2
issuing direct NFS commands under DOS: 2-6..2-8, 6-2
issuing direct NFS commands under PASCAL: 7-3

J

K

keyword: 5-2

L

LIST: 5-16
 list directory program: 8-13
 LOAD: 5-5
 LOCK: 4-10, 5-19..5-20, 5-35..5-37
 lockname: 4-10, 5-19..5-20, 5-35..5-37
 locks: 4-1
 locks versus passwords: 4-1
 lowercase: 8-6

M

/MAIN: 2-3, 5-16..5-17
 MINICMD: 7-11
 minimal system: 2-1
 minimum hardware requirements: 1-4
 MOUNT: 2-2, 4-6, 5-21..5-22, 5-38..5-40
 multiple file servers: 7-12

N

name: 3-5
 NESTAR.START: 8-4
 NESTED: 5-16..5-17
 NETLINK: 8-11
 NETUNIT: 7-9
 NFSCMD: 7-9
 NFSCMD1: 7-9
 NFSCMD0: 7-10

O

OFF: 5-23
 optional parameter: 5-2
 overview: 1-1

P

parameters: 5-2, E-3
 Pascal: 7-1..7-4
 Pascal with the Cluster/One System: 7-1..7-14
 Password: 3-5, 3-6, 4-3
 pathname: 3-3..3-6, 5-24
 POKE: 6-1
 PR#n: 2-5, 2-10, 2-11, 6-2, 6-3
 preset passwords: 4-7
 private access rights: 4-2..4-6
 profile: 8-3, 8-15
 PROTECT: 4-4, 5-25..5-26
 protection: 4-1..4-9, 5-9..5-10, 5-25..5-27, 5-41
 PRVACC: 4-4..4-5, 5-10, 5-25..5-27, 5-41

INDEX

PRVPW: 4-4..4-5, 5-10, 5-25..5-27, 5-41
PUBACC: 4-4..4-5, 5-10, 5-25..5-27, 5-41
public access rights: 4-2..4-5

Q

quick list of commands and parameters: E-1..E-4

R

reader comment form: ii
real diskettes: 6-6, 7-2, 7-5
Reference list: endpaper
reference NFSCMD0: 7-9
reference NFSCMD1: 7-10
RENAME: 5-28
REPLACE: 5-5..5-6
root directory: 2-2, 3-1..3-2
RO: 5-21
RW: 5-21
RWECD: 4-3..4-4

S

sector: 5-9, 6-1
SECTORS: 5-9, 6-1
security: 4-1
SET DIR: 5-29..5-30
SET DRIVE: 7-5
SET GRPPW: 5-32
SET NFS: 7-6
SET password: 5-32
SETPROFILE: 8-3 (Pascal), 8-15 (DOS)
SET PRVPW: 5-32
shared usage: 4-9, 5-11, 5-21
SHOW DATE: 5-33
SHOW DIR: 5-34
SHOW DRIVES: 7-4
SHOW LOCK: 4-11, 5-35
SHOW [ALL] LOCKS: 4-11, 5-36
SHOW [ALL] MOUNTS: 5-38
SHOW NFS: 7-5
SHOW PROTECTION: 4-6, 5-41
SHOW SLOTS: 7-5
SHOW STATION: 5-42
SHOW TYPE: 5-43
SHOQ TYPES: 5-44
SHOW VOLS: 5-45
SHR: 4-9, 5-11, 5-21
simplest disk organization: 2-2
SIZE: 5-6, 5-9, 6-1, 6-4, 7-1
slot: 1-4, 6-2, 7-5
Startcmd.Text: 8-4

STATION: 5-42
station number: D-1..D-2
syntax: 5-1
SYSTEM.APPLE: B-4
SYSTEM.MISCINFO: B-4
SYSTEM.PASCAL: B-4
SYSTEM.STARTUP: B-4

T

TDELETE: 8-10
TIMESTAMP: 5-46
tutorial for a minimal system: 2-1..2-14
type: 3-2, 5-9, 5-43

U

unit: 3-1, 5-45
unitname: 3-1, 3-3..3-4
UNLOCK: 4-11, 5-47
UNMOUNT: 5-48
usage: 4-1, 5-11, 5-21
user-classes: 4-2..4-3
user-stations: 1-4
utilities: 8-1..8-21

V

variable size DOS volumes: 6-1, 6-4
variable size Pascal volumes: 7-1
VDOS CREATE: 6-4
VERBOSE: 5-16
virtual diskettes: 2-1, 3-1
virtual diskettes of variable size: 6-4, 7-1
virtual drives: 2-2
volume: 3-2, 5-45

W

warranty: i

X

Y

Z

@

@: 2-5, 5-1, 6-2

CLUSTER/ONE MODEL A
NESTAR SYSTEMS, INCORPORATED

You are encouraged to use this form to communicate to Nestar any problems, bugs or suggestions associated with Nestar products. We would like your comments on improving the system itself, as well as this documentation.

Nestar installation location:

Submitter's name :

Address :

Nestar environment

Model A or Model One:

Hard disk size (if present):

Number of user stations:

Language used:

Type of application:

Description of problem or suggestion:

Please mail this form to:

Nestar Systems, Incorporated
2585 E. Bayshore
Palo Alto, California, 94303

REFERENCE FORM

COMMONLY USED VOLUME/FILE NAMES

Network Pathname	Pascal Volume Name	Pascal/DOS File Name	Size	Access/Protection
---------------------	-----------------------	-------------------------	------	-------------------

NESTAR SYSTEMS
INCORPORATED

*2585 East Bayshore Road
Palo Alto, California 94303*

USER'S MANUAL
Network File Server Version 1
Product Number A-2801
Licensed Publication Number LC20-0101-1

Printed in U.S.A.